

26  
Lej.



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

Facultad de Ingeniería

## El Análisis Cinemático de un Hexápodo y su Aplicación para Controlar el Espejo Secundario de un Telescopio

T E S I S  
Que para obtener el título de  
INGENIERO MECANICO ELECTRICISTA  
AREA MECANICA

p r e s e n t a

FLAVIO FIRMANI TULLI



Director de Tesis: M. en C. José Luis Pérez Silva

México, D. F. 1999

TESIS CON  
FALLA DE ORIGEN

271166



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

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

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

## *Agradecimientos*

*A mis padres:*

*Por el apoyo y dedicación que me han brindado a lo largo de la vida, y en especial para poder realizar este hermoso proyecto.*

*A José Luis:*

*Por la oportunidad que me ofreció de llevar a cabo este proyecto de Tesis, así como la motivación y generosidad que me proporcionó a lo largo de este tiempo.*

*A mi Alma Mater*

*Facultad de Ingeniería:*

*Por haberme albergado e instruido con dedicación y compromiso.*

*A mis profesores:*

*Por la formación académica y amistad que me otorgaron a lo largo de mi carrera universitaria.*

*Al Centro de Instrumentos:*

*Por la oportunidad que me dieron de poder realizar este proyecto.*

*Al Instituto de Astronomía:*

*Por la valiosa información que me proporcionaron para culminar este trabajo, en particular a Elfego Ruíz.*

*A Araceli:*

*Por el cariño y apoyo que a lo largo de esta carrera me dedicó, sin importar los sacrificios y las presiones que vivimos para poder cumplir nuestras metas y realizar varios sueños. De igual forma agradezco el apoyo que me brindó toda su familia.*

*A mis Abuelas:*

*A pesar de la lejanía, su cariño ha sido un aliciente para superarme en la vida.*

*A mis compañeros y amigos de la Facultad:*

*Por la amistad que me ofrecieron para poder alcanzar este objetivo.*

*A Lore:*

*Por el cariño que me dio a lo largo de la vida y su generosidad constante.*

*A mis amigos:*

*Por el afecto que me han manifestado y que ha servido de impulso para continuar por este camino.*

# ÍNDICE

<b>1. INTRODUCCIÓN</b> .....	1
1.1 Definición. ....	1
1.2 Descripción del mecanismo .....	2
1.3 Clasificación .....	3
1.4 Evolución Tecnológica.....	4
1.5 Ventajas y Desventajas .....	5
1.6 Aplicaciones .....	5
<b>2. OBJETIVO</b> .....	9
Propuesta .....	9
Selección del proyecto .....	10
<b>3. CINEMÁTICA</b> .....	11
3.1 Geometría General.....	12
3.1.1 Sistemas de Referencia.....	12
3.1.2 La posición de los vértices con respecto al centro de cada triángulo. ....	13
3.2 Cinemática Directa.....	15
3.2.1 Propuesta .....	15
3.2.2 Análisis para determinar la posición de los vértices.....	15
3.2.3 Análisis del vértice 1.....	16
3.2.4 Análisis del vértice 2.....	19
3.2.5 Análisis del vértice 3.....	21
3.2.6 Generación de los vectores de posición de los vértices.....	23
3.2.7 Control geométrico.....	24

3.2.8	<i>La posición correspondiente del vértice 3 en función del vértice 1.</i>	25
3.2.9	<i>La posición correspondiente del vértice 2 en función del vértice 1.</i>	30
3.2.10	<i>Comprobación de la distancia entre el vértice 2 y el vértice 3.</i>	32
3.2.11	<i>Matriz de Transformación.</i>	33
	<i>Vector de posición.</i>	33
	<i>Matriz de rotación.</i>	34
3.2.12	<i>Interpretación de la matriz de rotación</i>	36
	<i>Matriz de Euler.</i>	37
	<i>Matriz con vector auxiliar</i>	38
3.3	<i>Cinemática Inversa</i>	39
3.3.1	<i>Propuesta</i>	39
3.3.2	<i>Matriz de rotación de Euler</i>	40
3.3.3	<i>Matriz de rotación con vector auxiliar</i>	42
3.4	<i>Generación de trayectorias</i>	43
<b>4.</b>	<b><i>DINÁMICA</i></b>	<b>44</b>
4.1	<i>Desplazamientos Virtuales</i>	45
4.1.1	<i>Análisis matemático</i>	45
<b>5.</b>	<b><i>CONTROL DEL ESPEJO SECUNDARIO DE UN TELESCOPIO</i></b>	<b>47</b>
5.1	<i>Introducción</i>	47
5.1.1	<i>Clases de Telescopios</i>	48
	<i>Telescopio de Cassegrain y análogos</i>	51
5.1.2	<i>Movimientos de un telescopio</i>	52
5.1.3	<i>Monturas del telescopio</i>	53
5.2	<i>Descripción del problema.</i>	54
5.3	<i>Descripción de la propuesta.</i>	56
5.3.1	<i>Los seis grados de libertad del espejo</i>	57
5.3.2	<i>Sistema Real (Automatización)</i>	58
5.4	<i>Simulación.</i>	59
5.4.1	<i>Referencia</i>	60
5.4.2	<i>Distancia del lente de la cámara</i>	62
5.4.3	<i>Proyección de los haces</i>	64

5.5 Control Algoritmico. ....	66
5.5.1 Definiciones .....	66
5.5.2 Propuestas de control .....	67
5.6 Triángulo de referencia. ....	68
5.6.1 Desplazamiento XY.....	70
Análisis matemático del Desplazamiento XY. ....	71
5.6.2 Rotación .....	73
Análisis matemático de la Rotación en Z.....	75
5.6.3 Enfoque .....	76
Análisis matemático del Enfoque .....	77
5.6.4 Desplazamiento de los vértices .....	78
Análisis matemático del desplazamiento de los vértices. ....	79
5.6.5 Análisis del Algoritmo Raíz (Triángulo de Referencia).....	83
Media Pesada.....	84
Condición de salida.....	84
5.6.6 Diagrama de Bloques.....	85
5.7 Circunferencia de Referencia .....	86
5.7.1 Algoritmos secundarios I.....	87
5.7.2 Algoritmo primario.....	89
5.7.3 Algoritmos secundarios II.....	90
5.7.4 Análisis del Algoritmo Raíz (Circunferencia de Referencia) .....	92
5.7.5 Diagrama de Bloque .....	92
5.8 Triángulo de referencia con prueba láser.....	93
5.8.1 Prueba láser .....	93
5.8.2 Análisis de la propuesta .....	94
5.8.3 Diagrama de Bloques.....	95
<b>6. RESULTADOS Y COMPROBACIONES .....</b>	<b>96</b>
6.1 Cinemática Directa.....	97
6.2 Cinemática Inversa .....	99
6.3 Generación de trayectorias.....	100
6.4 Dinámica.....	101
6.4.1 Desplazamientos virtuales .....	101
6.4.2 Análisis estático.....	102
6.4.3 Desplazamiento de todos los actuadores.....	103

6.5 Control.....	104
<i>Presencia de los Algoritmos.....</i>	104
<i>Resultados de los programas de las propuestas de control.....</i>	105
<i>Triángulo de Referencia.....</i>	106
<i>Circunferencia de Referencia.....</i>	109
<i>Triángulo de Referencia con Láser.....</i>	112
<b>7. CONCLUSIONES.....</b>	<b>115</b>
<i>Cinemática.....</i>	115
<i>Dinámica.....</i>	116
<i>Prueba óptica.....</i>	117
<i>Propuestas de control.....</i>	117
<i>Consideraciones en los programas de computadora. ...</i>	118
<i>Resultados del Control.....</i>	119
<b>APÉNDICES.....</b>	<b>120</b>
<i>APÉNDICE A (CINEMÁTICA INVERSA).....</i>	120
<i>APÉNDICE B (BASES DE DATOS).....</i>	121
<i>APÉNDICE C (PROGRAMAS).....</i>	127
<i>Cinemática Directa.....</i>	127
<i>Cinemática Inversa (Euler).....</i>	137
<i>Cinemática Inversa (Vector Auxiliar).....</i>	139
<i>Dinámica.....</i>	142
<i>Control (Triángulo de Referencia).....</i>	144
<i>Control (Circunferencia de Referencia).....</i>	147
<i>Control (Triángulo de Referencia).....</i>	150
<b>REFERENCIAS.....</b>	<b>154</b>
<b>BIBLIOGRAFÍA DE APOYO.....</b>	<b>155</b>
<b>PUBLICACIONES.....</b>	<b>155</b>

# 1. INTRODUCCIÓN

## 1.1 Definición.

La palabra *hexápodo*, como su nombre lo indica, se refiere a un organismo u objeto de seis patas [(hexa) seis – (podo) patas]. En principio esta definición forma parte de un término de la biología (los artrópodos de seis patas o insectos), sin embargo en el mundo de la robótica este nombre se ha adaptado a los robots que presentan esta característica (seis patas). Existen hexápodos que tratan de simular a los insectos, y su principal finalidad es la de ser robots-caminadores. En este trabajo el hexápodo representa un manipulador de 6 DOF (grados de libertad) debido a los seis actuadores que trabajan en forma independiente. Por su forma de ocho caras triangulares también se le conoce como hexápodo octaédrico.

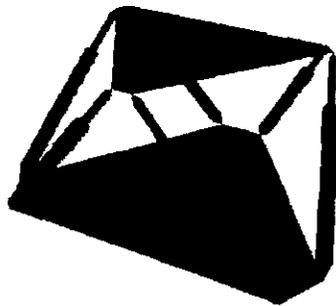


Figura 1.1 Hexápodo octaédrico

Un hexápodo, como se verá más adelante, es reconocido más por su precisión y estabilidad estructural, que por el espacio de trabajo que pueda alcanzar.

## 1.2 Descripción del mecanismo

Un hexápodo octaédrico es un mecanismo paralelo. Esta definición se debe a la topología que presenta, ya que está compuesto de dos plataformas que se encuentran conectadas mediante un sistema prismático de "patas" o actuadores que "actúan" en forma paralela. Las plataformas están definidas como: la plataforma base o fija que mantendrá su posición original sin importar las extensiones que pueden presentar los actuadores, y la plataforma móvil que variará su posición y orientación en presencia de estos cambios de longitud. Las dos plataformas representan los eslabones del manipulador. La unión entre cada actuador y las plataformas se llama articulación y es de la forma esférica o rotulada, mientras que el actuador representa otra articulación, de tipo prismático.

El extremo final de un robot se denomina órgano terminal. Este concepto se puede apreciar de manera clara en un robot con los eslabones conectados en serie, el órgano terminal será el último de los eslabones. Utilizando este mismo criterio para el caso del hexápodo, el órgano terminal será la plataforma móvil.

Cada uno de los movimientos independientes que puede realizar cada articulación constituye un **grado de libertad**. El número de grados de libertad viene dado por la suma de los grados de libertad de las articulaciones que lo componen y determina la accesibilidad de éste y su capacidad para orientar el órgano terminal. [ 1 ]

Con esta definición se pueden analizar los grados de libertad que posee el hexápodo. Las articulaciones de tipo esférico entre las plataformas y los actuadores no actúan en forma independiente y no introducen grados de libertad. Los únicos movimientos independientes se deben a los seis movimientos prismáticos de los actuadores, que se traducen en los 6 grados de libertad. Esto permite alcanzar la cantidad de variables necesarias para definir a un sólido en el espacio. Estas variables definen la posición y orientación que posee la plataforma móvil. La posición indica la localización del centro de la plataforma con respecto a los ejes de un sistema de referencia (X, Y, Z), mientras que la orientación está dada por los ángulos de rotación ( $\psi, \theta, \phi$ ) alrededor de los ejes del mismo sistema.

Aunque el movimiento prismático de cada actuador no depende de las longitudes de los demás actuadores, tenemos que tomar en cuenta una restricción. En efecto si se analizan las ocho caras triangulares del hexápodo, se puede apreciar que dos caras corresponden a las plataformas, mientras que cada una de las otras seis caras estará definida por un lado de la plataforma y por los actuadores. Debido a que la longitud de los actuadores es variable y que las caras son triangulares, no puede

haber un lado mayor que la suma de los otros dos. Esta restricción introduce una limitación en el espacio de trabajo del hexápodo.

### 1.3 Clasificación

Más allá del diseño conceptual del hexápodo, las necesidades han originado nuevos diseños, permitiendo una clasificación de hexápodos. Las diferencias en esta clasificación son los puntos donde están ubicadas las articulaciones esféricas. En la figura 1.2 se pueden apreciar ejemplos de esta clasificación, y el nombre que recibe cada hexápodo está acompañado de la configuración de las articulaciones en cada plataforma. Haciendo una visualización de esta diferencia, se concluye que al separar la posición de los actuadores sobre las plataformas se alcanza un mejor manejo en cuanto a éstos (no hay interferencias entre ellos a la hora de efectuar algún movimiento), pero se reduce el espacio de trabajo (workspace). Es importante definir este concepto porque al hacer el análisis matemático del manipulador existen diferencias de cálculo entre uno y otro.

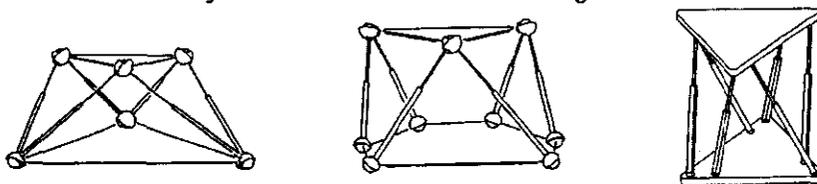


Figura 1.2 a) Hexápodo 3-3, b) hexápodo 6-3, c) hexápodo especial 6-6

Otra clasificación que se puede dar (aunque no interfiere en el cálculo matemático), es el tipo de actuador y articulación que se emplea. Por ejemplo pueden ser actuadores mecánicos (tornillos sin fin con motores de pasos), neumáticos (pistones neumáticos), tirantes (cuerdas en tensión); de igual forma se pueden encontrar articulaciones esféricas tipo cardan, o actuadores que cambian estructuralmente su orientación como los tirantes.

## 1.4 Evolución Tecnológica

El primer desarrollo que se tiene del hexápodo se remonta a 1949. La idea fue presentada por V. Gough como un mecanismo para hacer pruebas sobre llantas (Figura 1.3). Por esta razón el hexápodo también es conocido como la Plataforma de Gough. [ 2 ]

No obstante la persona que es considerada como el principal exponente en el desarrollo del hexápodo es D. Stewart, quien en 1965 escribió el artículo "A Platform with six degrees of freedom"; en él propone la idea de aplicar este mecanismo como el manipulador de un simulador de vuelo. Al mismo tiempo hace notar que este nuevo mecanismo se puede utilizar como la base para desarrollar nuevos tipos de máquinas herramientas [ 3 ]. Gracias a su trabajo, en el mundo científico el hexápodo es reconocido como "La Plataforma de Stewart" (Figura 1.4).

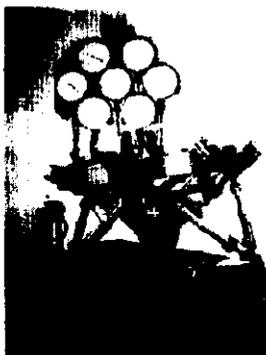


Figura 1.3 Plataforma de Gough 1949

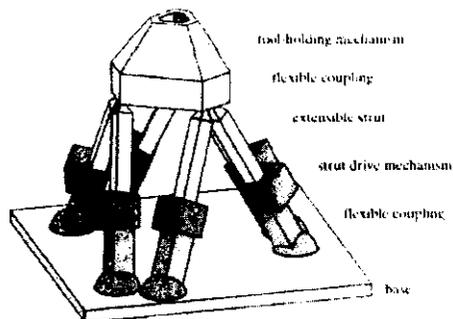


Figura 1.4 Plataforma de Stewart 1965

Las investigaciones sobre el hexápodo han continuado, especialmente esta década. Entre los investigadores más reconocidos se encuentran Merlet, J. P., Gosselin, C.M., Griffiths, M.W., Duffy, J., Kumar, V., Ji, Z., Lebert, G. La mayor parte de los artículos publicados tratan de estimar el espacio de trabajo que alcanza el manipulador. [ 1 ]

En las nuevas investigaciones, el ETH de Zurich ha desarrollado el Hexaglide, que es una variante del hexápodo y presenta la ventaja de abarcar un mayor espacio de trabajo. Esta ventaja se debe a que posee en las articulaciones esféricas, entre los actuadores y la plataforma fija, un movimiento prismático. Este movimiento se realiza a lo largo de rieles, como se ve en la figura 1.8.

### **1.5 Ventajas y Desventajas**

Entre las muchas ventajas que presenta el hexápodo, se tiene que:

- Es un manipulador de 6 grados de libertad
- Estructuralmente ocupa muy poco espacio para alcanzar 6 DOF
- Alcanza precisiones de hasta 1 micrómetro
- Los manipuladores con 6 DOF que tienen los mecanismos en serie, tienen el problema que los eslabones tienen que soportar las cargas en todas las direcciones, además que cada eslabón debe soportar el peso de los anteriores, mientras que en el caso de los mecanismos paralelos, como el hexápodo, solo tienen una carga en la dirección del actuador, por lo que solo deben soportar cargas de tracción o de compresión.

Entre las desventajas:

- El espacio de trabajo es reducido.
- Es muy difícil construir actuadores muy largos, ya que deben ser rígidos y precisos.

### **1.6 Aplicaciones**

A continuación se muestran algunas fotografías de aplicaciones en donde se emplea el hexápodo:

Una de las primeras aplicaciones que se realizaron con el hexápodo es el de ser el manipulador que controla los movimientos de las cabinas de entrenamiento de los simuladores de vuelo (Figura 1.5 y 1.6).

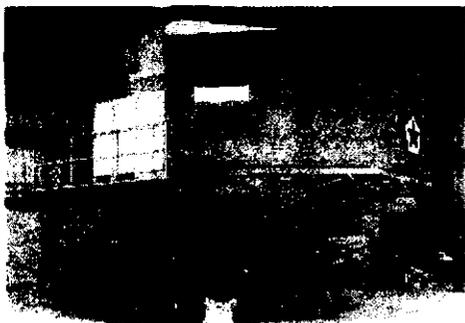
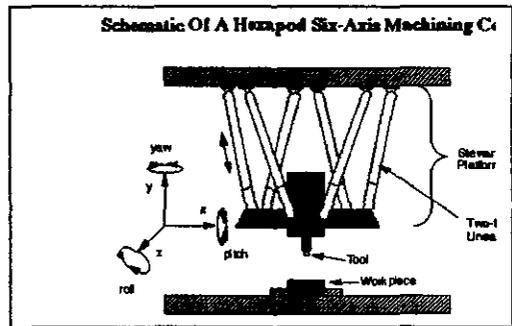


Figura 1.5 Simulador de vuelo MBB BO105



Figura 1.6 Simulador de vuelo UH-60

Otra aplicación importantísima, porque revoluciona el campo de las máquinas herramientas, es la de las nuevas fresadoras (Figuras 1.7).



Figuras 1.7 a) y b) Fresadora INGERSOLL, c) Simulación de la Fresadora INGERSOLL  
d) Esquema de una fresadora con 6 DOF

El Hexaglide, ya tiene sus primeras aplicaciones en el campo de las fresadoras, reiterando la ventaja de ser un manipulador que proporciona un mayor espacio de trabajo (Figura 1.8).

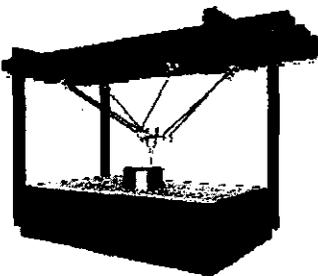


Figura 1.8 IWF Hexaglide

Una de las aplicaciones más sorprendentes que se pueden lograr con los hexápodos es la concatenación de éstos, para poder construir una serpiente mecánica con toda la flexibilidad que se desea. En la figura 1.9 se puede apreciar LOGABEX un robot diseñado para trabajar en una planta nuclear.

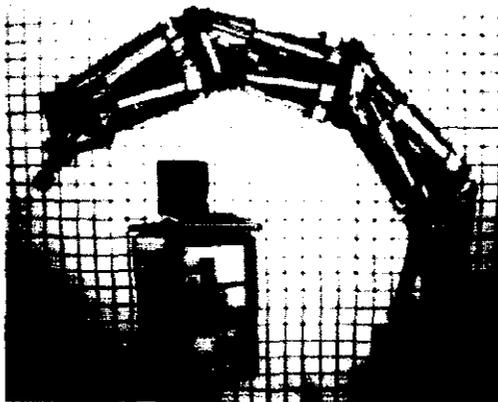


Figura 1.9 Manipulador LOGABEX

Entre las curiosidades está la simulación de un caballo. Este robot fue desarrollado en la Ecole Nationale d'Equitation de Francia (Figura 1.10).



Figura 1.10 Robot ecuestre

Una de las ventajas que tiene el hexápodo es la de ser un mecanismo que presenta una gran estabilidad y que alcanza de igual forma una alta precisión. Esto da lugar a que se pueda aplicar en una mesa de granito (Figura 1.11), permitiendo una precisión de 1 micrómetro y una carga de 500 kg. El nombre de este robot es ESRF. Otra aplicación del hexápodo es como manipulador de una antena de microondas (Figura 1.12).



Figura 1.11 El robot ESRF



Figura 1.12 Manipulador de una antena

El hexápodo se puede utilizar también en sistemas ópticos, por ejemplo en la difracción de rayos X, como posicionador de aparatos ópticos, láser, cañones de electrones y como el manipulador que controla al espejo secundario de telescopios de alta resolución (Figura 1.13).

Figura 1.13 Telescopio infrarrojo UKIRT, realizado por el Royal Observatory de Edinburgh y el Instituto Max-Planck de Heidelberg.



## 2. OBJETIVO

*La importancia del hexápodo radica en el impacto tecnológico que presenta en el campo de la robótica. El hexápodo representa uno de los mecanismos más completos y eficientes ya que, a pesar del reducido espacio que ocupa, tiene la capacidad de emplear seis grados de libertad. No obstante su sencillez estructural, el desarrollo de este sistema es reciente y ha traído consigo mucho éxito. Este auge ha generado que se convierta en una alternativa para varios sistemas mecánicos.*

### **Propuesta**

*Como se puede desprender del título de la Tesis, el proyecto está dividido en dos trabajos de investigación, "El análisis cinemático de un hexápodo" y "su aplicación para controlar el espejo secundario de un telescopio".*

**El análisis cinemático de un hexápodo** representa el estudio del movimiento del hexápodo. Existen dos conceptos fundamentales dentro del análisis cinemático, la Cinemática Directa y la Cinemática Inversa. La primera permite determinar cual es la posición y orientación del órgano terminal del robot, con respecto a un sistema de referencia dado, para lo cual es necesario conocer la longitud de cada uno de los actuadores. En el caso de la cinemática inversa se puede determinar la longitud de cada actuador en función de la orientación y posición del órgano terminal del manipulador.

*A partir de este análisis se pueden crear trayectorias. La cinemática permite, dada una posición inicial, llegar a una posición final, y obtener las nuevas longitudes que deben alcanzar los actuadores.*

*Una aplicación del hexápodo, **controlar el espejo secundario de un telescopio**, es la segunda parte del proyecto. En ella se busca aplicar este manipulador a un problema real, que es el de mantener en el mismo eje focal a los dos espejos, primario y secundario, de un telescopio. Como se verá más adelante la dificultad empieza en el momento que se desconoce la posición del espejo secundario, entonces mediante pruebas ópticas se establecerá el control de los 6 grados de libertad del hexápodo, mediante la proyección de imágenes y el uso de un adecuado algoritmo de control.*

*Esta aplicación se puede desarrollar en dos formas: una es la aplicación real y en ella es necesario construir el manipulador y aplicar el sistema óptico en un telescopio, la otra opción es la de simular el sistema mediante un software.*

### **Selección del proyecto**

*En la introducción se hizo referencia a una clasificación entre los hexápodos, y dependiendo de la elección había diferencias en el análisis matemático de éste. Este proyecto está planteado para el desarrollo de un hexápodo 3-3, es decir con tres puntos de unión entre las plataformas y los actuadores.*

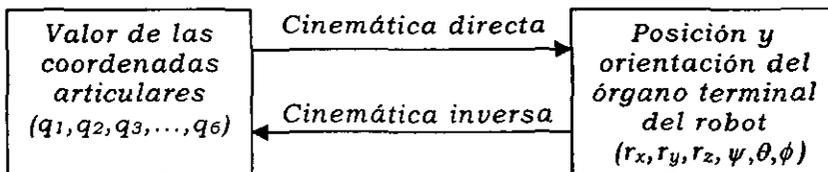
*De igual forma sobre la base de una clasificación entre los diferentes tipos de actuadores, se eligieron los actuadores que contienen el sistema de control longitudinal, ya que como se verá más adelante para desarrollar la cinemática directa es necesario conocer cual es la longitud de cada actuador.*

*Sobre las dos opciones mencionadas para la aplicación del hexápodo como el manipulador que controla el espejo secundario de un telescopio, se optó por la simulación mediante un software, ya que ésta permite comprobar la eficiencia del sistema de control sin tener que invertir económicamente en el proyecto.*

### 3. CINEMÁTICA

La cinemática de un robot estudia su comportamiento con respecto a un sistema de referencia; se obtiene una descripción analítica del movimiento espacial del robot, mediante el empleo del álgebra vectorial y matricial. [ 4 ]

Existen dos conceptos fundamentales dentro de la cinemática, la Cinemática Directa y la Cinemática Inversa. La primera permite determinar cual es la posición y orientación del extremo final del robot, u órgano terminal, con respecto a un sistema de referencia dado; para esto es necesario conocer las características geométricas de las articulaciones, ya sean los desplazamientos o las rotaciones que puedan tener. En el caso de la Cinemática Inversa se pueden obtener las características geométricas de las articulaciones mediante la posición y orientación final del manipulador. Por lo que se tiene la siguiente relación:



En el caso del hexápodo, las coordenadas articulares representan el desplazamiento de cada actuador ( $L_i$ ), ya que son los únicos movimientos independientes. Por otro lado los parámetros geométricos del órgano terminal están dados por la

posición del centro de la plataforma móvil ( $r_x, r_y, r_z$ ) y la orientación que presenta dicha plataforma ( $\psi, \theta, \phi$ ), con respecto a un sistema de referencia fijo.

### 3.1 Geometría General

Para poder desarrollar la cinemática es necesario generar los sistemas de referencia que definen al hexápodo y, en base a éstos, se puede determinar cuales son las posiciones que tienen los vértices de las plataformas.

#### 3.1.1 Sistemas de Referencia.

Ambas plataformas del hexápodo (fija y móvil) son dos triángulos equiláteros, de manera que el centro de cada triángulo será el origen de un sistema de referencia. El triángulo fijo tendrá el sistema de referencia ( $O$ ) cuyos ejes son ( $x, y, z$ ) y los vértices  $P_4, P_5, P_6$ ; mientras que en el centro del triángulo móvil estará el sistema de referencia ( $O^*$ ), de ejes ( $x^*, y^*, z^*$ ) y como vértices a  $P_1, P_2, P_3$  (Figura 3.1).

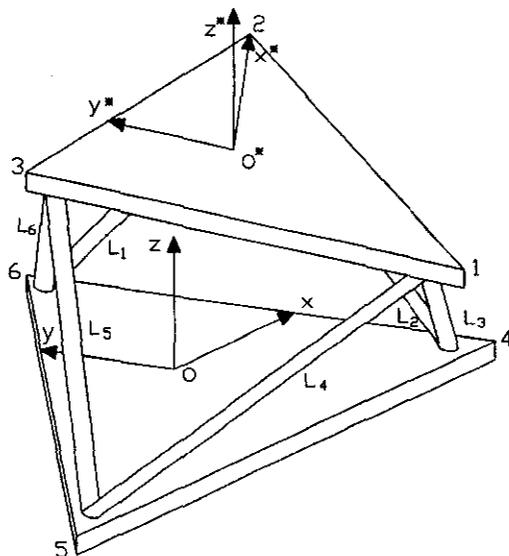


Figura 3.1 Sistemas de Referencia

### 3.1.2 La posición de los vértices con respecto al centro de cada triángulo.

Definidos los dos sistemas de referencia, se puede determinar la posición de los vértices con respecto al centro de cada triángulo.

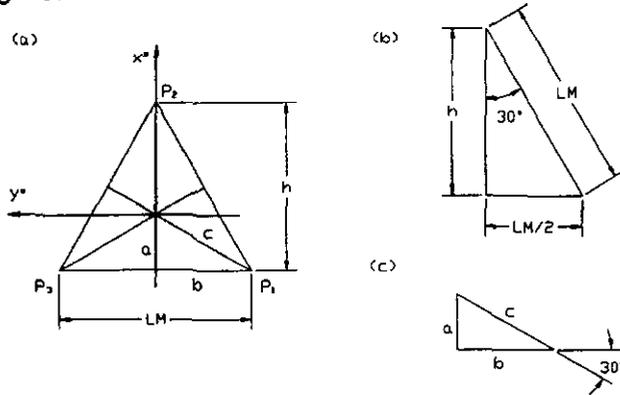


Figura 3.2 a) Plataforma móvil, b) Triángulo de lados  $h$ - $LM$ - $LM/2$ ,  
c) Triángulo de lados  $abc$

De la figura 3.2b se puede deducir que:

$$h = LM \cos 30^\circ = \frac{\sqrt{3}}{2} LM \quad (3.1)$$

Por otro lado de la figura 3.2c resulta, por el teorema de Pitágoras, que:

$$a^2 + b^2 = c^2 \quad (3.2)$$

Además:

$$b = \frac{LM}{2} \quad (3.3)$$

$$a + c = h = \frac{\sqrt{3}}{2} LM \quad (3.4)$$

Sustituyendo las ecuaciones (3.3) y (3.4) en la (3.2):

$$a^2 + \frac{LM^2}{4} = \left( \frac{\sqrt{3}}{2} LM - a \right)^2$$

Simplificando

$$a = \frac{LM}{2\sqrt{3}} \quad (3.5)$$

Sustituyendo la ecuación (3.5) en la (3.3) resulta

$$c = \frac{\sqrt{3}}{2} LM - \frac{LM}{2\sqrt{3}} \quad c = \frac{LM}{\sqrt{3}} \quad (3.6)$$

Por lo tanto las coordenadas de los vértices del triángulo móvil son:

$$\begin{array}{l} P_1 \left( -\frac{LM}{2\sqrt{3}}, -\frac{LM}{2} \right) \\ P_2 \left( \frac{LM}{\sqrt{3}}, 0 \right) \\ P_3 \left( -\frac{LM}{2\sqrt{3}}, \frac{LM}{2} \right) \end{array}$$

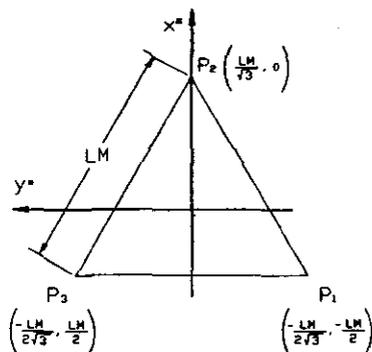


Figura 3.3 Triángulo móvil

De forma análoga se determinan las coordenadas de los vértices del triángulo fijo.

$$\begin{array}{l} P_4 \left( \frac{LF}{2\sqrt{3}}, -\frac{LF}{2} \right) \\ P_5 \left( -\frac{LF}{\sqrt{3}}, 0 \right) \\ P_6 \left( \frac{LF}{2\sqrt{3}}, \frac{LF}{2} \right) \end{array}$$

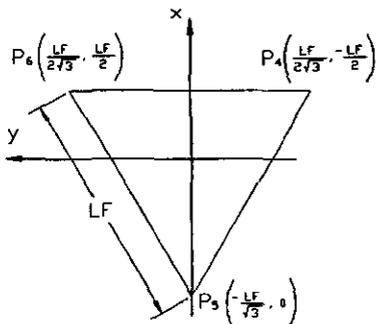


Figura 3.4 Triángulo fijo.

### 3.2 Cinemática Directa

La Cinemática Directa en un hexápodo determina la posición y orientación final del centro del triángulo móvil como función de las longitudes de los actuadores.

$$x^* = f(L_1, L_2, L_3, L_4, L_5, L_6)$$

$$y^* = f(L_1, L_2, L_3, L_4, L_5, L_6)$$

$$z^* = f(L_1, L_2, L_3, L_4, L_5, L_6)$$

$$\psi = f(L_1, L_2, L_3, L_4, L_5, L_6); \text{ con respecto al eje X}$$

$$\theta = f(L_1, L_2, L_3, L_4, L_5, L_6); \text{ con respecto al eje Y}$$

$$\phi = f(L_1, L_2, L_3, L_4, L_5, L_6); \text{ con respecto al eje Z}$$

La Cinemática Directa permite referir el sistema  $O^*$ , con respecto al sistema  $O$ , para cualquier configuración de longitudes que presenten los actuadores. La relación entre ambos sistemas estará dada por una matriz de transformación, así que el objetivo de este capítulo será determinar dicha matriz en función de las longitudes de los actuadores.

#### 3.2.1 Propuesta

La propuesta que se presenta en este proyecto para resolver la Cinemática Directa es la siguiente: definidas las longitudes de los actuadores, hay que determinar cual es la única posible posición que pueden tener los vértices del triángulo móvil con respecto al sistema  $O$ , para dichas longitudes; posteriormente, se genera la matriz de transformación en función de las posiciones que alcanzaron los vértices.

#### 3.2.2 Análisis para determinar la posición de los vértices.

Una primera visualización de los posibles puntos por los cuales pueden estar los vértices se puede hacer de la siguiente manera: debido a que se conoce la longitud de los actuadores, así como los lados de cada uno de los triángulos del hexápodo, se puede observar que los dos actuadores que se intersectan en un vértice del triángulo móvil y el lado del triángulo fijo que une a los dos actuadores, forman otro triángulo. Analizando este triángulo se puede ver que el único lado que se encuentra determinado vectorialmente es el lado del triángulo fijo. Si se

definen dos vectores, uno como el lado del triángulo fijo y otro como la altura del triángulo que se visualizó, se puede realizar el siguiente análisis: si se rota el vector altura, alrededor del vector que define al lado del triángulo fijo, se generará una circunferencia, la cual representará las posibles posiciones que puede tener el vértice (Figura 3.7). Donde la altura será el radio y la intersección entre los dos vectores será el centro de la circunferencia.

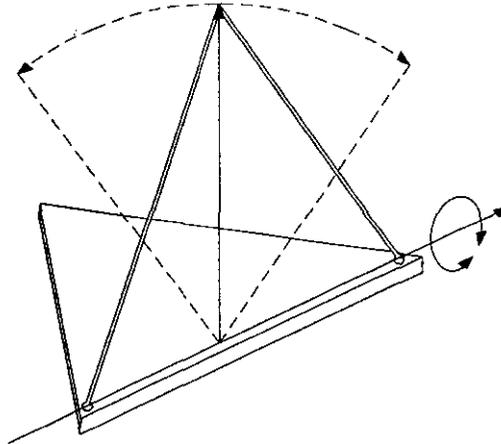


Figura 3.5 Análisis de los posibles puntos donde puede estar el vértice.

### 3.2.3 Análisis del vértice 1.

Para analizar el vértice 1 se emplea la propuesta de generar la circunferencia que representa a los puntos donde puede encontrarse el vértice, en función de los actuadores  $L_3$  y  $L_4$ , como se ve en la figura 3.6.

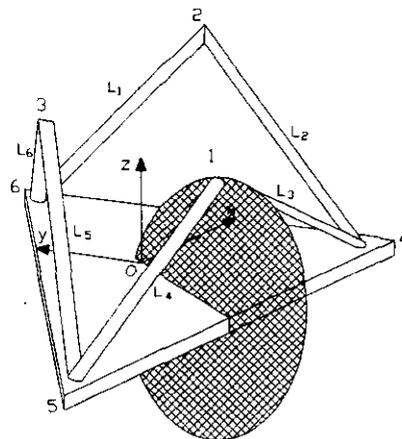


Figura 3.6 Análisis del vértice 1.

Se puede observar que el radio de la circunferencia es la altura del triángulo generado por los puntos 1,4,5, como se ve en la figura 3.7.

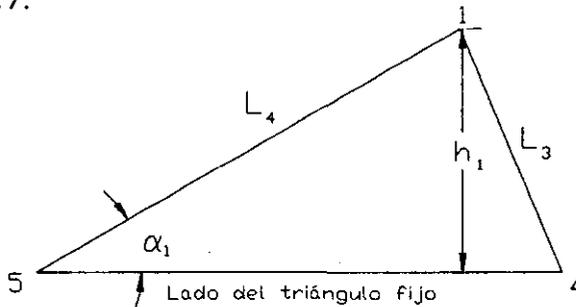


Figura 3.7 Triángulo 1-4-5.

Para encontrar el valor del ángulo  $\alpha_1$ , se emplea la ley de cosenos, dada por la siguiente expresión:

$$\alpha_1 = \cos^{-1} \left( \frac{LF^2 + L_4^2 - L_3^2}{2(LF)(L_4)} \right)$$

Con el valor de  $\alpha_1$ , se obtiene  $h_1$ , o sea el radio de la circunferencia.

$$r_1 = h_1 = L_4 \text{ sen}(\alpha_1)$$

Mientras que el centro de la circunferencia ( $C_1$ ) se determina mediante un vector de posición ( $\vec{V}_1$ ), como se ve en la figura 3.8, en la proyección en el plano XY.

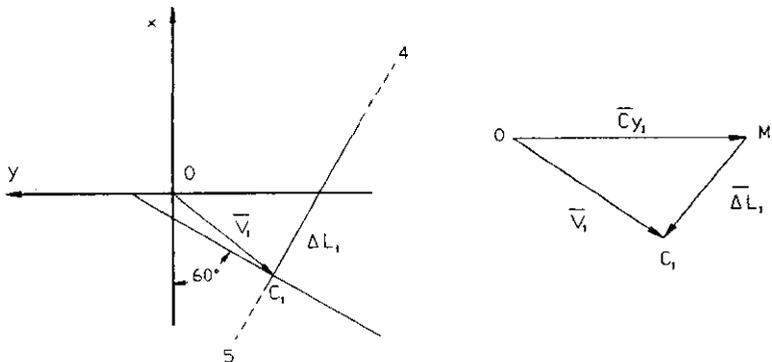
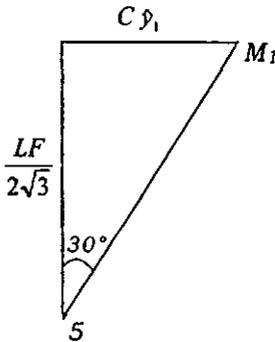


Figura 3.8. Proyección de la circunferencia 1 en el plano XY

El vector  $\hat{V}_1$  se determina de la figura 3.8, por suma vectorial.

$$\hat{V}_1 = \Delta L_1 + C \hat{y}_1 \quad (3.7)$$

Retomando la figura 3.4 se puede representar el detalle que permite obtener el valor de  $C \hat{y}_1$ .



Debido a que

$$\tan 30^\circ = \frac{-C \hat{y}_1}{\frac{LF}{\sqrt{3}}}$$

y que  $\tan 30^\circ = \frac{1}{\sqrt{3}}$ ; se tiene

$$C \hat{y}_1 = -\left(\frac{1}{\sqrt{3}}\right) \cdot \frac{LF}{\sqrt{3}} = -\frac{LF}{3} \hat{j} \quad (3.8)$$

Detalle de la Figura 3.4

Por otro lado se obtiene  $\Delta L_1$ , o sea el vector que va del centro de la circunferencia ( $C_1$ ), al punto  $M_1$ . Se determina como la resta entre el vector  $\overline{C_1 P_5}$  y el vector  $\overline{M_1 P_5}$ . El vector  $\overline{C_1 P_5}$  se obtiene con la proyección de  $L_4$  sobre el triángulo fijo, (Figura 3.7); mientras que sus componentes se determinan por el ángulo de  $60^\circ$  entre  $C \hat{y}_1$  y  $\Delta L_1$  (Figura 3.8). La proyección de  $L_4$  está dada por:

$$\text{Proy}_{LF} L_4 = L_4 \cos(\alpha_1)$$

Por lo que

$$\Delta L_1 = \left( \frac{\sqrt{3}}{2} \cdot (L_4 \cos(\alpha_1)) - \frac{LF}{\sqrt{3}}, \frac{1}{2} \cdot (L_4 \cos(\alpha_1)) + \frac{LF}{3}, 0 \right) \quad (3.9)$$

Finalmente las componentes del vector  $\hat{V}_1$  se obtienen sustituyendo las ecuaciones (3.8) y (3.9) en la (3.7):

$$\hat{V}_1 = \left( \frac{\sqrt{3}}{2} \cdot (L_4 \cos(\alpha_1)) - \frac{LF}{\sqrt{3}}, \frac{1}{2} \cdot (L_4 \cos(\alpha_1)) + \frac{LF}{3}, 0 \right) \quad (3.10)$$

### 3.2.4 Análisis del vértice 2.

Para analizar el vértice 2 se desarrolla el mismo criterio de la circunferencia que define a los posibles puntos en que puede estar el vértice 2, en este caso los actuadores son  $L_1$  y  $L_2$ .

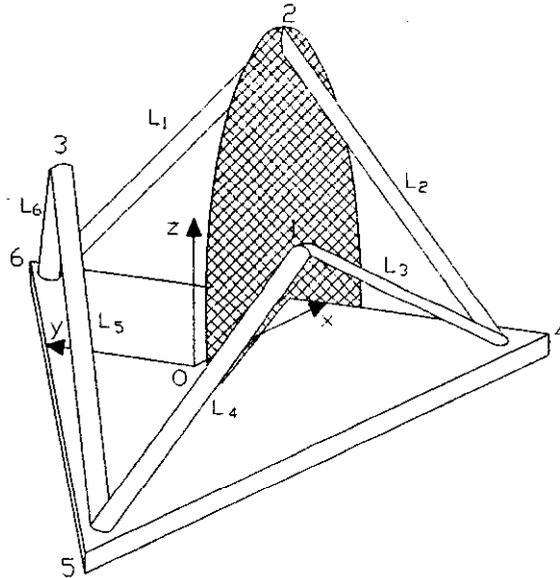


Figura 3.9 Análisis del vértice 2.

De la misma forma el triángulo que se genera tiene como altura el radio de la circunferencia y está formado por los puntos 2, 4, 6.

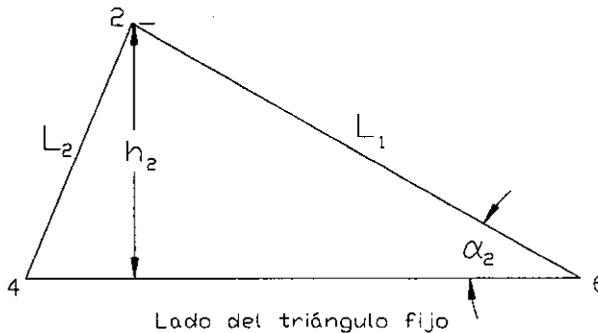


Figura 3.10 Triángulo 2-4-6.

De manera análoga se utiliza la ley de cosenos para encontrar el valor del ángulo  $\alpha_2$ .

$$\alpha_2 = \cos^{-1} \left( \frac{LF^2 + L_1^2 - L_2^2}{2(LF)(L_1)} \right)$$

Con el valor de  $\alpha_2$  se obtiene la altura ( $h_2$ ) del triángulo 2,4,6, o sea el radio de la circunferencia ( $r_2$ ).

$$r_2 = h_2 = L_1 \text{ sen}(\alpha_2)$$

Para obtener el vector de posición del centro de la circunferencia ( $\hat{V}_2$ ), se analiza la proyección en el plano XY.

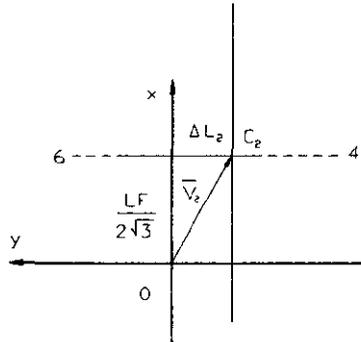


Figura 3.11 Proyección de la circunferencia 2 en el plano XY

Empleando la suma de vectores

$$\hat{V}_2 = \Delta L_2 + \frac{LF}{2\sqrt{3}} i \quad (3.11)$$

Donde  $\Delta L_2$  está dado por

$$\Delta L_2 = \left( 0, \frac{LF}{2} - (L_1 \cos(\alpha_2)), 0 \right) \quad (3.12)$$

Finalmente las componentes del vector  $\hat{V}_2$  se obtienen sustituyendo las ecuaciones (3.12) en (3.11):

$$\hat{V}_2 = \left( \frac{LF}{2\sqrt{3}}, \frac{LF}{2} - L_1 \cos(\alpha_2), 0 \right) \quad (3.13)$$

### 3.2.5 Análisis del vértice 3.

El análisis del vértice 3 se desarrolla en forma análoga al realizado en el análisis del vértice 1. Solo que en este caso los actuadores son  $L_5$  y  $L_6$ .

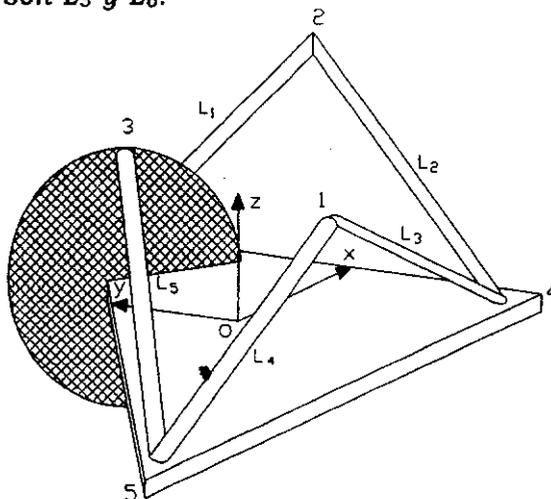


Figura 3.12 Análisis del vértice 3.

De la misma forma el triángulo que se genera tiene como altura el radio de la circunferencia y en este caso pasa por los puntos 3,5,6 (Figura 3.13).

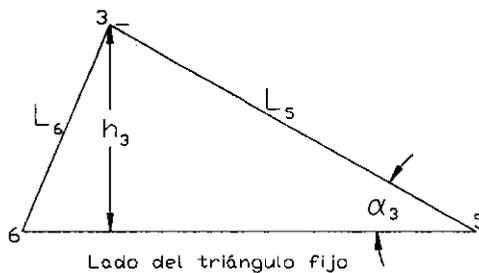


Figura 3.13 Triángulo 3-5-6.

De manera análoga se utiliza la ley de cosenos para encontrar el valor del ángulo  $\alpha_3$ .

$$\alpha_3 = \cos^{-1} \left( \frac{LF^2 + L_5 - L_6}{2(LF)(L_5)} \right)$$

Con el valor de  $\alpha_3$ , se obtiene  $h_3$  ó  $r_3$ .

$$r_3 = h_3 = L_3 \text{ sen}(\alpha_3)$$

Para obtener el vector de posición ( $\vec{V}_3$ ) del centro de la circunferencia, se analiza la proyección en el plano XY.

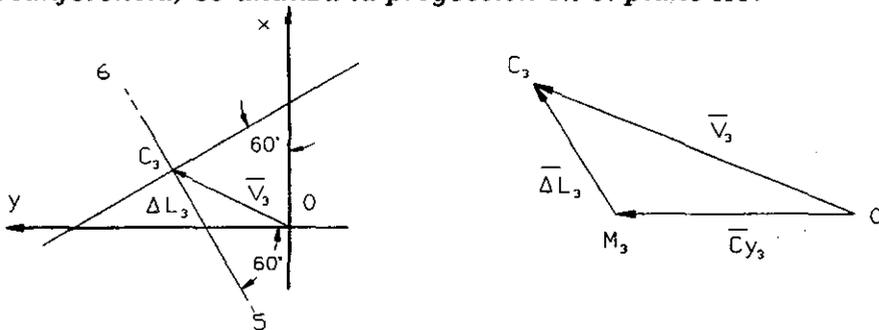


Figura 3.14 Proyección de la circunferencia 3 en el plano XY

El vector  $\vec{V}_3$ , se determina en forma análoga al vector  $\vec{V}_1$ .

$$\vec{V}_3 = \Delta L_3 + C y_3 \quad (3.14)$$

donde

$$C y_3 = -C y_1 \quad (3.15)$$

Mientras que  $\Delta L_3$ , se desarrolla como la resta entre el vector  $\overline{C_3 P_3}$  y el vector  $\overline{M_3 P_3}$ . La proyección de  $L_3$  está dada por:

$$\text{Proy}_{LF} L_3 = L_3 \cos(\alpha_1)$$

Por lo que  $\Delta L_3$ , quedaría como:

$$\Delta L_3 = \left( \frac{\sqrt{3}}{2} \cdot (L_3 \cos(\alpha_3)) - \frac{LF}{\sqrt{3}}, \frac{1}{2} \cdot (L_3 \cos(\alpha_3)) - \frac{LF}{3}, 0 \right) \quad (3.16)$$

Finalmente las componentes del vector  $\vec{V}_3$ , se obtienen sustituyendo las ecuaciones (3.15) y (3.16) en (3.14):

$$\vec{V}_3 = \left( \frac{\sqrt{3}}{2} \cdot (L_3 \cos(\alpha_3)) - \frac{LF}{\sqrt{3}}, \frac{1}{2} \cdot (L_3 \cos(\alpha_3)) - \frac{LF}{3}, 0 \right) \quad (3.17)$$

### 3.2.6 Generación de los vectores de posición de los vértices

La generación del vector de posición de cada vértice se obtiene mediante la matriz de transformación  ${}^0T_{C_i}$ , por lo que resulta la siguiente expresión:

$${}^0P_i = {}^0T_{C_i} {}^C_iP_i$$

donde  $i = 1, 2, 3$

Con esta expresión se traslada el sistema de referencia al centro de cada circunferencia. Además los nuevos sistemas se orientarán de manera que el plano XZ sea el plano de la circunferencia, y  $\phi_i$  el ángulo que permite esta orientación. Los puntos que conforman la circunferencia se determinarán al variar el ángulo  $\theta_i$  correspondiente (Figura 3.15).

$$\begin{pmatrix} Px_i \\ Py_i \\ Pz_i \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \phi_i & -\text{sen } \phi_i & 0 & Vx_i \\ \text{sen } \phi_i & \cos \phi_i & 0 & Vy_i \\ 0 & 0 & 1 & Vz_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} |r_i| \cdot \cos \theta_i \\ 0 \\ |r_i| \cdot \text{sen } \theta_i \\ 1 \end{pmatrix} = \begin{pmatrix} |r_i| \cdot \cos \theta_i \cdot \cos \phi_i + Vx_i \\ |r_i| \cdot \cos \theta_i \cdot \text{sen } \phi_i + Vy_i \\ |r_i| \cdot \text{sen } \theta_i + Vz_i \\ 1 \end{pmatrix} \quad (3.18)$$

donde  $\phi_1 = 60^\circ$ ,  $\phi_2 = 180^\circ$ ,  $\phi_3 = 300^\circ$   
 $i = 1, 2, 3$

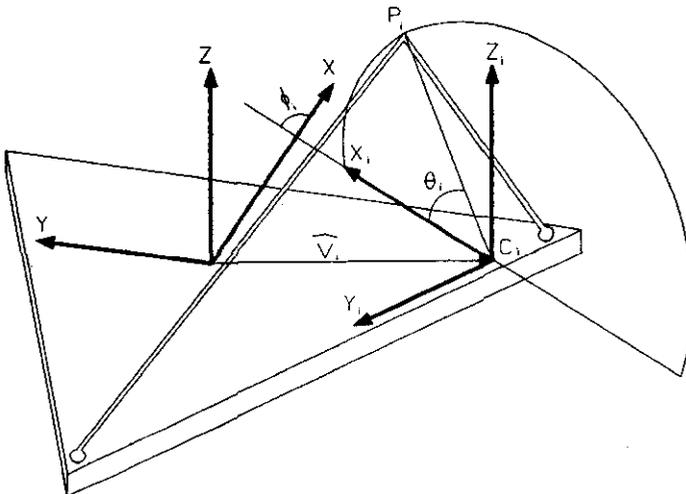


Figura 3.15 Representación vectorial de los vértices

### **3.2.7 Control geométrico.**

*El hecho de que sólo exista una posición de los vértices para una cierta configuración longitudinal de los actuadores, da como resultado la necesidad de aplicar un control, llamado control geométrico. La condición de comparación y/o salida que se utiliza para determinar si se alcanzó la posición de los vértices que satisface a dichas longitudes, es la distancia entre cada uno de los vértices. Esta distancia representa la longitud de cada lado del triángulo móvil y las tres distancias entre vértices son iguales, por ser un triángulo equilátero.*

*En el caso que no exista alguna posición de los vértices que cumpla con una cierta configuración de los actuadores, se considerará que el sistema diverge; si en cambio se alcanza una solución, entonces el sistema convergerá y habrá una y solamente una posible posición de los vértices.*

*Para desarrollar el control geométrico se divide el problema en tres etapas:*

- 1) Suponiendo una posición del vértice 1, se determina la posición correspondiente del vértice 3, tal que la distancia entre ambos puntos sea la longitud del lado del triángulo móvil (LM).*
- 2) De forma análoga se determina la posición del vértice 2, a partir de la misma posición que tiene el vértice 1, bajo la misma condición que la distancia entre ambos puntos sea LM.*
- 3) Finalmente se comprueba si la distancia entre el vértice 2 y el vértice 3 es LM; si no se cumple esta condición, se supone otro valor de la posición del vértice 1 y se repite el ciclo, hasta obtener la convergencia.*

*Las tres etapas representan, cada una, una comparación entre vértices, bajo la condición que la distancia entre ellos sea igual a LM. A continuación se analiza cada una de estas etapas.*

### 3.2.8 La posición correspondiente del vértice 3 en función del vértice 1.

El primer paso es fijar la posición del vértice 1, para esto se define un valor del ángulo  $\theta_1$  (como se ve en la Figura 3.15); con esta condición se realiza un análisis de cuáles son los vectores que partiendo del vértice 1 alcanzan a la circunferencia que describe los posibles puntos del vértice 3; estos vectores generan geoméricamente un cono (Figura 3.16), y el problema se reduce en encontrar el vector cuya distancia sea igual a  $LM$ .

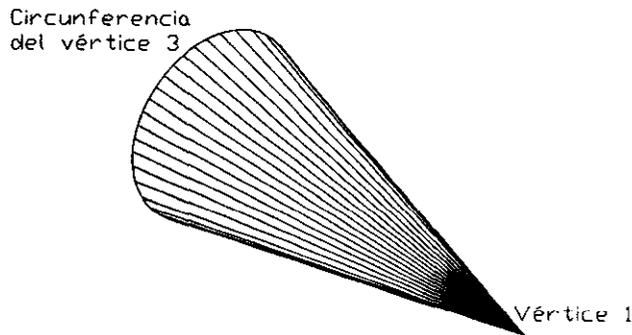


Figura 3.16 El cono que definen los vectores que parten del vértice 1 y alcanzan a la circunferencia 3.

Para esto es necesario conocer los límites máximo y mínimo, que se determinan de la siguiente manera:

Si la longitud  $LM$  es menor que la distancia mínima que existe entre el punto y la circunferencia, el sistema diverge. De igual forma el sistema también diverge si la longitud  $LM$  es mayor que la distancia máxima.

Por lo tanto se puede concluir que para que el sistema converja debe satisfacer la siguiente expresión:

$$L_{13,min} \leq LM \leq L_{13,max}$$

Estos límites se pueden determinar de diferentes formas. Una manera podría ser creando un plano que contenga al vértice 1 y al vector que pasa por los puntos 5 y 6; esto implicará que los puntos donde intersecta el plano a la circunferencia del vértice 3 representen los límites máximo y mínimo.

Otra forma para obtener estos límites es la siguiente: si se proyecta al vértice 1 en el plano  $XY$  y se determina el vector de mínima distancia ( $d_1$ ) entre esta proyección y la recta que pasa por los puntos 5 y 6 ( $L_3$ ), se puede definir el ángulo ( $\phi_{13}$ ), como se ve en la figura 3.17.

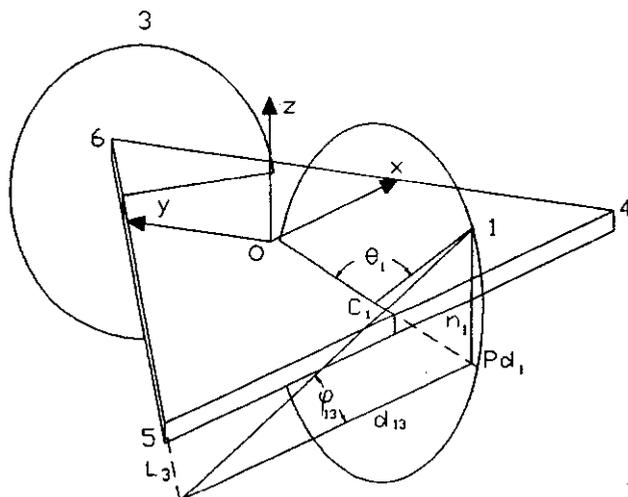


Figura 3.17 Distancias máxima y mínima entre el vértice 1 y la circunferencia 3.

En forma algebraica este desarrollo se expresa de la siguiente manera:

$$\phi_{13} = \tan^{-1} \left( \frac{n_1}{d_{13}} \right) \quad (3.19)$$

$$\text{donde } n_1 = r_1 \sin \theta_1 \\ d_{13} = |d_{13}|$$

El punto  $Pd_1$  se expresa como:

$$Pd_1 = \begin{pmatrix} \cos \phi_1 & -\sin \phi_1 & 0 & Vx_1 \\ \sin \phi_1 & \cos \phi_1 & 0 & Vy_1 \\ 0 & 0 & 1 & Vz_1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \|r_1\| \cdot \cos \theta_1 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \|r_1\| \cdot \cos \theta_1 \cdot \cos \phi_1 + Vx_1 \\ \|r_1\| \cdot \cos \theta_1 \cdot \sin \phi_1 + Vy_1 \\ Vz_1 \\ 1 \end{pmatrix} \quad (3.20)$$

El valor de  $Pd_1$  de la ecuación (3.20) se sustituye en la ecuación de la mínima distancia:

$$d_{13} = \frac{\|(Pd_1 - P_6) \times (P_5 - P_6)\|}{\|(P_5 - P_6)\|} \quad (3.21)$$

Finalmente se sustituye el resultado de la ecuación (3.21) en la ecuación (3.19) y se obtiene el valor de  $\varphi_{13}$ .

Si se traslada el ángulo  $\varphi_{13}$  al plano de la circunferencia se determina el vector de mínima distancia entre el vértice 1 y la circunferencia 3. Las coordenadas de dicho punto se obtienen al sustituir  $\theta_3 = \varphi_{13}$  en la ecuación (3.18)

$$P_{3min} = \begin{pmatrix} \cos \phi_3 & -\text{sen } \phi_3 & 0 & Vx_3 \\ \text{sen } \phi_3 & \cos \phi_3 & 0 & Vy_3 \\ 0 & 0 & 1 & Vz_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \|r_3\| \cdot \cos \varphi_{13} \\ 0 \\ \|r_3\| \cdot \text{sen } \varphi_{13} \\ 1 \end{pmatrix} = \begin{pmatrix} \|r_3\| \cdot \cos \varphi_{13} \cdot \cos \phi_3 + Vx_3 \\ \|r_3\| \cdot \cos \varphi_{13} \cdot \text{sen } \phi_3 + Vy_3 \\ \|r_3\| \cdot \text{sen } \varphi_{13} + Vz_3 \\ 1 \end{pmatrix}$$

Las coordenadas del punto de la circunferencia 3, donde el vector que parte del vértice 1 alcanza su máxima distancia, se obtienen con el mismo criterio, ya que  $P_{3max}$  es diametralmente opuesto a  $P_{3min}$ , así que el valor de  $\theta_3$ , que se sustituye en la ecuación (3.18) es  $\varphi_{13} + 180^\circ$ .

$$P_{3max} = \begin{pmatrix} \cos \phi_3 & -\text{sen } \phi_3 & 0 & Vx_3 \\ \text{sen } \phi_3 & \cos \phi_3 & 0 & Vy_3 \\ 0 & 0 & 1 & Vz_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \|r_3\| \cdot \cos(\varphi_{13} + 180) \\ 0 \\ \|r_3\| \cdot \text{sen}(\varphi_{13} + 180) \\ 1 \end{pmatrix} = \begin{pmatrix} \|r_3\| \cdot \cos(\varphi_{13} + 180) \cdot \cos \phi_3 + Vx_3 \\ \|r_3\| \cdot \cos(\varphi_{13} + 180) \cdot \text{sen } \phi_3 + Vy_3 \\ \|r_3\| \cdot \text{sen}(\varphi_{13} + 180) + Vz_3 \\ 1 \end{pmatrix}$$

Finalmente la mínima y máxima distancia entre el vértice 1 y la circunferencia del vértice 3 serán:

$$l_{13min} = \sqrt{(P_{x3min} - P_{x1})^2 + (P_{y3min} - P_{y1})^2 + (P_{z3min} - P_{z1})^2}$$

$$l_{13max} = \sqrt{(P_{x3max} - P_{x1})^2 + (P_{y3max} - P_{y1})^2 + (P_{z3max} - P_{z1})^2}$$

Una vez encontrados los puntos máximos y mínimos, se coloca un sistema de referencia en el centro de la circunferencia con la rotación del ángulo  $\varphi_{13}$  (Figura 3.18).

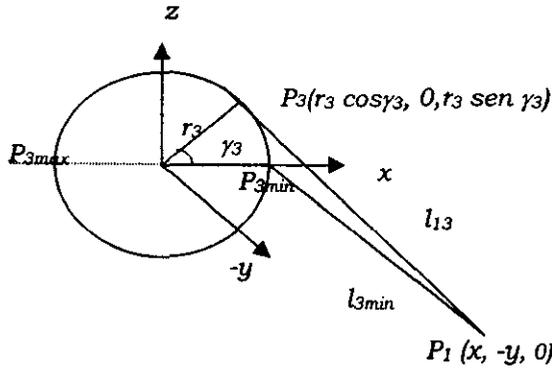


Figura 3.18 Proyección del plano de la circunferencia, con el sistema rotado para que los puntos límite estén en el eje X.

Esta nueva representación permite desarrollar la siguiente expresión algebraica.

$$\begin{aligned}
 l_{13}^2 &= ((x - r_3 \cos \gamma_3)^2 + y^2 + r_3^2 \sin^2 \gamma_3) \\
 &= x^2 - 2r_3 x \cos \gamma_3 + r_3^2 \cos^2 \gamma_3 + y^2 + r_3^2 \sin^2 \gamma_3 \\
 &= x^2 + y^2 + r_3^2 - 2r_3 x \cos \gamma_3
 \end{aligned} \tag{3.22}$$

Si  $\gamma_3 = 0^\circ$  entonces

$$l_{13min}^2 = x^2 + y^2 + r_3^2 - 2r_3 x$$

Si  $\gamma_3 = 180^\circ$  entonces

$$l_{13max}^2 = x^2 + y^2 + r_3^2 + 2r_3 x$$

Sumando estos dos términos se tiene que:

$$\frac{l_{13min}^2 + l_{13max}^2}{2} = x^2 + y^2 + r_3^2 \tag{3.23}$$

Si se restan

$$\frac{l_{13max}^2 - l_{13min}^2}{2} = 2r_3 x \tag{3.24}$$

Sustituyendo las ecuaciones (3.23) y (3.24) en la (3.22) se obtiene que:

$$l_{13}^2 = \frac{l_{13min}^2 + l_{13max}^2}{2} - \frac{l_{13max}^2 - l_{13min}^2}{2} \cos \gamma_3$$

En esta expresión la única incógnita es  $\gamma_3$ , ya que el valor de  $l_{13}$  es la longitud del triángulo móvil (LM).

$$\gamma_3 = \cos^{-1} \left( \frac{l_{13min}^2 + l_{13max}^2 - 2l_{13}^2}{l_{13max}^2 - l_{13min}^2} \right)$$

Por lo tanto la posición correspondiente del vértice 3 para una posición dada del vértice 1 presenta la siguiente conjetura: la distancia de un punto con respecto a una circunferencia tiene dos posibles soluciones, así que para un valor de  $\theta_1$ , habrá dos valores de  $\theta_3$ , (Figura 3.19). Matemáticamente estos dos valores se obtienen de la siguiente forma:

$$\theta_3 = \varphi_{13} + \gamma_3 \quad (3.25a)$$

$$\theta_3' = \varphi_{13} - \gamma_3 \quad (3.25b)$$

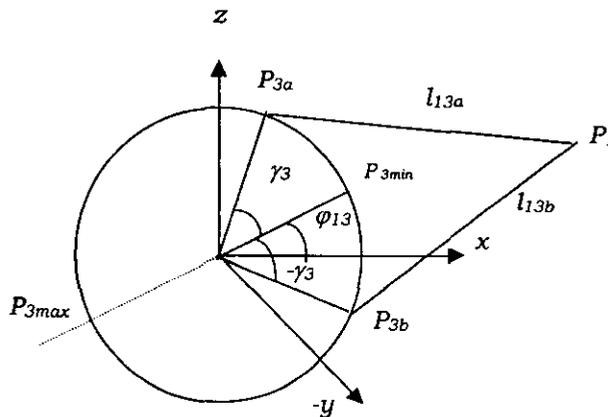


Figura 3.19 Las dos posiciones que puede tener el vértice 3.

### 3.2.9 La posición correspondiente del vértice 2 en función del vértice 1.

Si existe solución en la etapa anterior y se obtienen los dos valores del ángulo  $\theta_1$ , se puede desarrollar esta segunda etapa, que corresponde a la posición del vértice 2 en función del vértice 1, manteniendo el mismo valor del ángulo  $\theta_1$ . Por la similitud que presenta con la etapa anterior sólo se describirán las ecuaciones principales. En este caso el rango que permite la convergencia será:

$$L_{12} \min \leq LM \leq L_{12} \max$$

De la misma forma se realiza el análisis para determinar los límites, sólo que ahora se determina la mínima distancia que existe entre el vértice 1 proyectado en el plano XY y la recta ( $L_2$ ) que pasa por los puntos 4 y 6. Finalmente se obtiene el ángulo  $\varphi_{12}$ , como se ve en la figura 3.20.

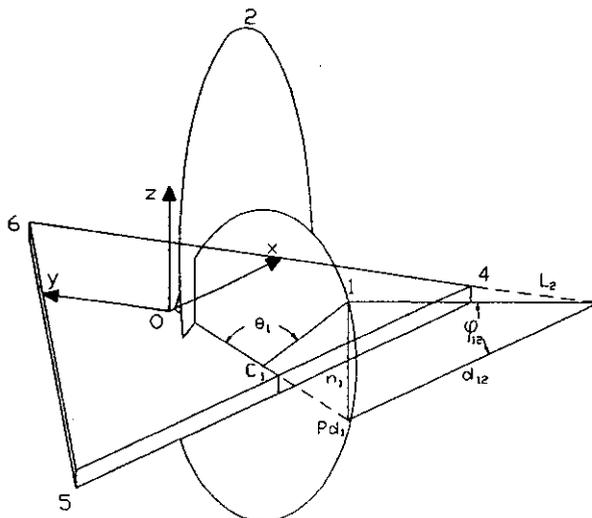


Figura 3.20 Distancias máxima y mínima entre el vértice 1 y la circunferencia 2.

Las ecuaciones se pueden escribir de la siguiente forma:

$$\varphi_{12} = \tan^{-1} \left( \frac{n_1}{d_{12}} \right) \quad (3.26)$$

donde  $n_1 = r_1 \sin \theta_1$   
 $d_{12} = \left\| \hat{d}_{12} \right\|$

La proyección del vértice 1 es la misma que en la etapa anterior, así que en la ecuación de la mínima distancia ( $d_{12}$ ) se sustituye el valor obtenido de  $Pd_1$  en la ecuación (3.20)

$$d_{12} = \frac{\| (Pd_1 - P_6) \times (P_4 - P_6) \|}{\| (P_4 - P_6) \|} \quad (3.27)$$

Se sustituye el valor obtenido de  $d_{12}$ , de la ecuación (3.27), en la ecuación (3.26). De esta forma con el valor de  $\varphi_{12}$  se determinan, sustituyendo en la ecuación (3.18), los puntos de la circunferencia 2 con los que el vector que parte del vértice 1 alcanza sus valores máximo y mínimo.

Mínimo  $\theta_2 = \varphi_{12}$

$$P_{2min} = \begin{pmatrix} \cos \phi_2 & -\text{sen } \phi_2 & 0 & Vx_2 \\ \text{sen } \phi_2 & \cos \phi_2 & 0 & Vy_2 \\ 0 & 0 & 1 & Vz_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \|r_2\| \cdot \cos \varphi_{12} \\ 0 \\ \|r_2\| \cdot \text{sen } \varphi_{12} \\ 1 \end{pmatrix} = \begin{pmatrix} \|r_2\| \cdot \cos \varphi_{12} \cdot \cos \phi_2 + Vx_2 \\ \|r_2\| \cdot \cos \varphi_{12} \cdot \text{sen } \phi_2 + Vy_2 \\ \|r_2\| \cdot \text{sen } \varphi_{12} + Vz_2 \\ 1 \end{pmatrix}$$

Máximo  $\theta_2 = \varphi_{12} + 180^\circ$

$$P_{2max} = \begin{pmatrix} \cos \phi_2 & -\text{sen } \phi_2 & 0 & Vx_2 \\ \text{sen } \phi_2 & \cos \phi_2 & 0 & Vy_2 \\ 0 & 0 & 1 & Vz_2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \|r_2\| \cdot \cos(\varphi_{12} + 180) \\ 0 \\ \|r_2\| \cdot \text{sen}(\varphi_{12} + 180) \\ 1 \end{pmatrix} = \begin{pmatrix} \|r_2\| \cdot \cos \varphi_{12} \cdot \cos(\phi_2 + 180) + Vx_2 \\ \|r_2\| \cdot \cos \varphi_{12} \cdot \text{sen}(\phi_2 + 180) + Vy_2 \\ \|r_2\| \cdot \text{sen}(\varphi_{12} + 180) + Vz_2 \\ 1 \end{pmatrix}$$

Las longitudes mínima y máxima entre el vértice 1 y la circunferencia del vértice 2 son:

$$l_{12min} = \sqrt{(P_{x2min} - P_{x1})^2 + (P_{y2min} - P_{y1})^2 + (P_{z2min} - P_{z1})^2}$$

$$l_{12max} = \sqrt{(P_{x2max} - P_{x1})^2 + (P_{y2max} - P_{y1})^2 + (P_{z2max} - P_{z1})^2}$$

Finalmente con el mismo criterio que en el caso anterior se obtiene que:

$$\gamma_2 = \cos^{-1} \left( \frac{l_{12\min}^2 + l_{12\max}^2 - 2l_{12}^2}{l_{12\max}^2 - l_{12\min}^2} \right) ; \text{ donde } l_{12} \text{ es LM}$$

En este caso las dos posibles soluciones serán:

$$\theta_2 = \varphi_{12} + \gamma_2 \quad (3.28a)$$

$$\theta_2' = \varphi_{12} - \gamma_2 \quad (3.28b)$$

### 3.2.10 Comprobación de la distancia entre el vértice 2 y el vértice 3.

Esta última etapa del control geométrico es la que permite comprobar si el ángulo  $\theta_1$  que se eligió fue el correcto. Para desarrollar esta etapa es necesario utilizar los dos posibles ángulos obtenidos en las etapas anteriores, y sustituirlos en la ecuación de la representación vectorial de los vértices (3.18), para así determinar los puntos  $P_{2a}$ ,  $P_{2b}$ ,  $P_{3a}$  y  $P_{3b}$ . Se sustituyen estos puntos en la ecuación (3.29), de tal manera que se combinen entre sí.

$$l_{23} = \sqrt{(P_{2xi} - P_{3xj})^2 + (P_{2yi} - P_{3yj})^2 + (P_{2zi} - P_{3zj})^2} \quad (3.29)$$

$$\begin{aligned} \text{donde } i &= a, b \\ j &= a, b \end{aligned}$$

En total habrá cuatro posibles soluciones, pero si converge el sistema, o sea ( $l_{23} = LM$ ), sólo se presentará para una configuración de puntos y para un solo valor particular de  $\theta_1$ . Por esta razón se desarrolló un programa con un algoritmo que permite iterar el valor de  $\theta_1$  hasta encontrar los valores correctos de los ángulos  $\theta_2$  y  $\theta_3$ . El ángulo  $\theta_1$  variará de 0 a 180° debido al espacio de trabajo permitido de los actuadores.

De esta forma se ha obtenido la única posición que pueden alcanzar los vértices para una cierta configuración longitudinal de los actuadores.

### 3.2.11 Matriz de Transformación.

En la Propuesta (sección 3.2.1) se describió que, después de obtener las posiciones de los vértices, había que determinar la matriz de transformación que define a los parámetros geométricos ( $r_x, r_y, r_z, \psi, \theta, \phi$ ).

Una matriz de transformación está compuesta por un vector de posición y una matriz de rotación, y permite referir el sistema  $O^*$  del sistema  $O$ .

#### Vector de posición

Con las componentes de los vectores de posición que definen a los vértices del triángulo móvil se puede determinar el vector de posición que describe al centro del mismo triángulo, o sea el origen del sistema  $O^*$ . Este vector de posición se determina con el promedio de las componentes de los tres vértices (Figura 3.21).

$$r = \left( \frac{(P_{1x} + P_{2x} + P_{3x})}{3}, \frac{(P_{1y} + P_{2y} + P_{3y})}{3}, \frac{(P_{1z} + P_{2z} + P_{3z})}{3} \right) \quad (3.30)$$

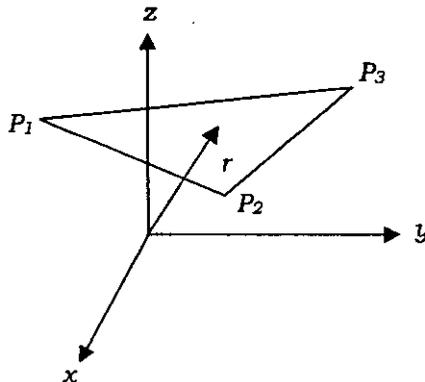


Figura 3.21 Vector de posición

### **Matriz de rotación**

Para determinar la matriz de rotación se traslada el triángulo móvil al fijo mediante el vector de posición.

$$\begin{aligned} P'_1 &= (P_{1x} - r_x, P_{1y} - r_y, P_{1z} - r_z) \\ P'_2 &= (P_{2x} - r_x, P_{2y} - r_y, P_{2z} - r_z) \\ P'_3 &= (P_{3x} - r_x, P_{3y} - r_y, P_{3z} - r_z) \end{aligned}$$

Cualquier matriz de rotación en  $R^3$  se expresa de la siguiente forma:

$${}^O R = \begin{pmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{pmatrix} \quad (3.31)$$

(el sistema  $O^*$  referido al sistema  $O$ , de la figura 3.1)

El problema se reduce en generar la matriz de rotación con los parámetros que se han obtenido.

La primera columna significa referir el eje  $x^*$  con respecto al sistema  $O$ ; la referencia resulta bastante sencilla ya que el eje  $x^*$  se dirige hacia al vértice 2 y se pueden emplear los cosenos directores

$$xx^* = \frac{P'_{2x}}{\|P'_2\|} \quad yx^* = \frac{P'_{2y}}{\|P'_2\|} \quad zx^* = \frac{P'_{2z}}{\|P'_2\|}$$

donde

$$\|P'_2\| = \sqrt{(P'_{2x})^2 + (P'_{2y})^2 + (P'_{2z})^2}$$

Para la segunda columna, la referencia del eje  $y^*$  con respecto al sistema  $O$ , se determina obteniendo el punto  $(C_{23})$  donde el eje  $y^*$  intersecta al triángulo móvil (Figura 3.22).

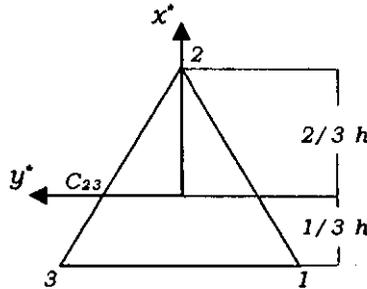


Figura 3.22 Intersección del eje  $y^*$  con el triángulo móvil

Debido a que el eje  $y^*$  cruza el lado que se encuentra entre los vértices 2 y 3 a una distancia de  $1/3$  más cercana del vértice 3, se puede definir que las componentes del vector de posición del punto intersección  $(C_{23})$  son:

$$C_{23} \left( \frac{2P'_{3x} + P'_{2x}}{3}, \frac{2P'_{3y} + P'_{2y}}{3}, \frac{2P'_{3z} + P'_{2z}}{3} \right)$$

Por lo que la segunda columna de la matriz quedará:

$$xy^* = \frac{C_{23x}}{\|C_{23}\|} \quad yy^* = \frac{C_{23y}}{\|C_{23}\|} \quad zy^* = \frac{C_{23z}}{\|C_{23}\|}$$

donde

$$\|C_{23}\| = \sqrt{(C_{23x})^2 + (C_{23y})^2 + (C_{23z})^2}$$

La última columna, la cual corresponde al eje  $z^*$ , se obtiene mediante el producto cruz de las dos columnas anteriores, ya que el vector del eje  $z^*$  es ortogonal a los ejes  $x^*$  y  $y^*$ .

$$K = \begin{bmatrix} K_x \\ K_y \\ K_z \end{bmatrix} = \begin{bmatrix} i & -j & k \\ P'_{1x} & P'_{2y} & P'_{2z} \\ C_{23x} & C_{23y} & C_{23z} \end{bmatrix}$$

Resolviendo:

$$K_x = (P'_{2y}C_{23z} - P'_{2z}C_{23y})$$

$$K_y = (P'_{2z}C_{23x} - P'_{2x}C_{23z})$$

$$K_z = (P'_{2x}C_{23y} - P'_{2y}C_{23x})$$

Finalmente:

$$xz^{\cdot} = \frac{K_x}{\|K\|} \quad yz^{\cdot} = \frac{K_y}{\|K\|} \quad zz^{\cdot} = \frac{K_z}{\|K\|}$$

donde

$$\|K\| = \sqrt{K_x^2 + K_y^2 + K_z^2}$$

Por lo tanto la matriz de rotación quedará formada de la siguiente manera:

$${}^0_oR = \begin{pmatrix} xx^{\cdot} & xy^{\cdot} & xz^{\cdot} \\ yx^{\cdot} & yy^{\cdot} & yz^{\cdot} \\ zx^{\cdot} & zy^{\cdot} & zz^{\cdot} \end{pmatrix} = \begin{bmatrix} \frac{P'_{2x}}{\|P'_2\|} & \frac{C_{23x}}{\|C_{23}\|} & \frac{K_x}{\|K\|} \\ \frac{P'_{2y}}{\|P'_2\|} & \frac{C_{23y}}{\|C_{23}\|} & \frac{K_y}{\|K\|} \\ \frac{P'_{2z}}{\|P'_2\|} & \frac{C_{23z}}{\|C_{23}\|} & \frac{K_z}{\|K\|} \end{bmatrix} \quad (3.32)$$

### 3.2.12 Interpretación de la matriz de rotación

La matriz que se generó está compuesta por valores numéricos. Para determinar los valores de los parámetros geométricos  $(\psi, \theta, \phi)$  se confronta esta matriz con alguna estándar. En este proyecto se consideraron dos matrices, la primera que permite determinar los ángulos de Euler  $(\psi, \theta, \phi)$ , mientras que la segunda es una matriz de rotación con un eje arbitrario o vector auxiliar.

### Matriz de Euler

La matriz de Euler [ 5 ] incluye tres ángulos, los cuales rotan alrededor de cada uno de los ejes del sistema O:

$\psi$  con respecto al eje x (desviación).

$\theta$  con respecto al eje y (elevación).

$\phi$  con respecto al eje z (giro).

$$R_{\phi,\theta,\psi} = R_{x,\phi} \cdot R_{y,\theta} \cdot R_{z,\psi} = \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix}$$

Igualando las dos matrices se tiene que:

De la celda (3,1)

$$-\text{sen}\theta = \frac{P'_{2z}}{\|P'_2\|}$$

despejando  $\theta$

$$\theta = \text{sen}^{-1} \left( -\frac{P'_{2z}}{\|P'_2\|} \right)$$

De la celda (2,1)

$$\text{sen}\phi \cos\theta = \frac{P'_{2y}}{\|P'_2\|}$$

despejando  $\phi$

$$\phi = \text{sen}^{-1} \left( \frac{1}{\cos\theta} \left( \frac{P'_{2y}}{\|P'_2\|} \right) \right)$$

De la celda (3,2)

$$\cos\theta \text{sen}\psi = \frac{C_{23z}}{\|C_{23}\|}$$

despejando  $\psi$

$$\psi = \text{sen}^{-1} \left( \frac{1}{\cos\theta} \left( \frac{C_{23z}}{\|C_{23}\|} \right) \right)$$

De esta forma se obtiene la matriz de rotación en función de los parámetros geométricos ( $\psi, \theta, \phi$ ). Finalmente la matriz de transformación quedaría expresada como:

$${}^o_o T = \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi & r_x \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi & r_y \\ -s\theta & c\theta s\psi & c\theta c\psi & r_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.33)$$

### Matriz con vector auxiliar

Este método consiste en crear un vector en el origen y generar una rotación alrededor del mismo de un ángulo  $\alpha$  [ 6 ], como se ve en la figura 3.23.

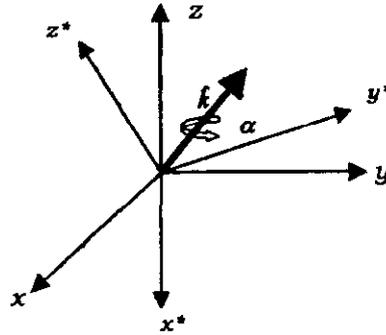


Figura 3.23 Vector auxiliar

La matriz tiene la siguiente forma

$$R_{k,\alpha} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = \begin{bmatrix} k_x^2 v\alpha + c\alpha & k_x k_y v\alpha - k_z s\alpha & k_x k_z v\alpha + k_y s\alpha \\ k_x k_y v\alpha + k_z s\alpha & k_y^2 v\alpha + c\alpha & k_y k_z v\alpha - k_x s\alpha \\ k_x k_z v\alpha - k_y s\alpha & k_y k_z v\alpha + k_x s\alpha & k_z^2 v\alpha + c\alpha \end{bmatrix}$$

$$\text{donde } v\alpha = \text{versa} = 1 - \text{cosa}$$

El ángulo de giro del vector auxiliar se obtiene de la siguiente manera:

$$\alpha = \cos^{-1} \left( \frac{r_{11} + r_{22} + r_{33} - 1}{2} \right) = \cos^{-1} \left( \frac{\frac{P'_{2x}}{|P'_2|} + \frac{C_{22y}}{|C_{23}|} + \frac{K_x}{|K|} - 1}{2} \right)$$

Mientras que las componentes del vector  $\hat{k}$  se determinan como:

$$\hat{k} = \frac{1}{2 \text{sen } \alpha} \begin{bmatrix} r_{32} - r_{23} \\ r_{13} - r_{31} \\ r_{21} - r_{12} \end{bmatrix} = \frac{1}{2 \text{sen } \alpha} \begin{bmatrix} \frac{C_{23z}}{|C_{23}|} - \frac{K_y}{|K|} \\ \frac{K_x}{|K|} - \frac{P'_{2z}}{|P'_2|} \\ \frac{P'_{2y}}{|P'_2|} - \frac{C_{23x}}{|C_{23}|} \end{bmatrix}$$

### 3.3 Cinemática Inversa

Con la Cinemática Inversa se puede obtener la longitud de cada uno de los actuadores en función de los parámetros geométricos ( $r_x, r_y, r_z, \psi, \theta, \phi$ ).

#### 3.3.1 Propuesta

Para resolver este problema se considera la siguiente observación: debido a que cada actuador se puede representar como un vector que une a un vértice del triángulo fijo con uno del móvil, se puede determinar la longitud de cada uno de los actuadores aplicando una suma vectorial. El primer vector parte del vértice del triángulo fijo donde se encuentra el actuador y se dirige al centro del mismo triángulo ( $O$ ); de ahí partirá otro vector, que representa el vector de posición entre los dos triángulos (el vector de posición  $r$  de la Cinemática Directa); finalmente un vector que se dirige al vértice del triángulo móvil, mismo vértice que alcanza el actuador en mención (Figura 3.24).

Para desarrollar esta suma vectorial se utilizan los dos sistemas de referencia empleados hasta el momento  $O$  y  $O^*$ , que tienen su origen en el centro de cada triángulo.

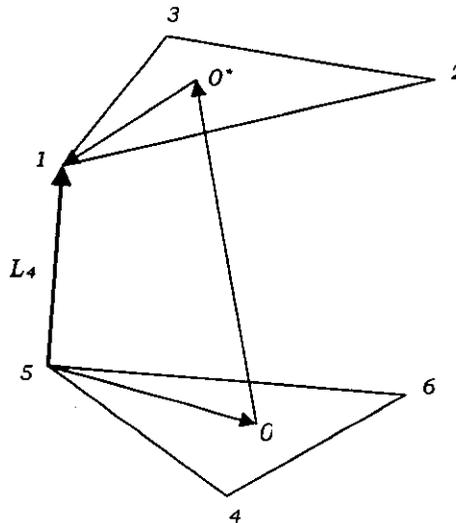


Figura 3.24 Suma Vectorial

Para desarrollar la Cinemática Inversa se van a considerar las dos matrices de rotación que se utilizaron en la Cinemática Directa; la que incluye los ángulos de Euler y la del vector auxiliar. De esta manera se podrá comprobar que los valores obtenidos en ambas cinemáticas son los correctos.

### 3.3.2 Matriz de rotación de Euler

En base a la propuesta anterior, las componentes del primer vector se obtienen como la diferencia entre las coordenadas de los vértices y el centro del sistema O:

$${}^4P_0 = \left( -\frac{LF}{2\sqrt{3}}, -\frac{LF}{2}, 0 \right) \quad {}^5P_0 = \left( -\frac{LF}{2\sqrt{3}}, \frac{LF}{2}, 0 \right) \quad {}^6P_0 = \left( -\frac{LF}{\sqrt{3}}, 0, 0 \right)$$

Por lo que respecta al segundo vector se utiliza el vector de posición  $(r_x, r_y, r_z)$  que se expuso en la Cinemática Directa; además, puesto que el tercer vector de la suma vectorial está referido al sistema  $O^*$ , se aplica la matriz de rotación. Conjuntando el vector de posición y la matriz de rotación se está describiendo a la matriz de transformación  ${}^0T$ :

$${}^0T = \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi & r_x \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi & r_y \\ -s\theta & c\theta s\psi & c\theta c\psi & r_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Finalmente el tercer vector de la suma vectorial se obtiene de manera análoga a la realizada para obtener el primer vector.

$${}^0P_1 = \left( -\frac{LM}{2\sqrt{3}}, -\frac{LM}{2}, 0 \right) \quad {}^0P_2 = \left( \frac{LM}{\sqrt{3}}, 0, 0 \right) \quad {}^0P_0 = \left( -\frac{LM}{2\sqrt{3}}, \frac{LM}{2}, 0 \right)$$

Las sumas vectoriales que determinan la longitud de cada actuador quedarán expresadas de la siguiente manera:

${}^6P_2 = {}^6T_0 {}^0T_0 {}^0P_2$  ; la magnitud de  ${}^6P_2$  representa el actuador  $L_1$

${}^4P_2 = {}^4T_0 {}^0T_0 {}^0P_2$  ; la magnitud de  ${}^4P_2$  representa el actuador  $L_2$

${}^4P_1 = {}^4T_0 {}^0T_0 {}^0P_1$  ; la magnitud de  ${}^4P_1$  representa el actuador  $L_3$

${}^5P_1 = {}^5T_0 {}^0T_0 {}^0P_1$  ; la magnitud de  ${}^5P_1$  representa el actuador  $L_4$

${}^5P_3 = {}^5T_0 {}^0T_0 {}^0P_3$  ; la magnitud de  ${}^5P_3$  representa el actuador  $L_5$

${}^6P_3 = {}^6T_0 {}^0T_0 {}^0P_3$  ; la magnitud de  ${}^6P_3$  representa el actuador  $L_6$

La matriz de transformación  ${}^{nf}T_0$  está compuesta por una matriz de rotación, que es una matriz identidad por no presentar rotación, y por el vector de posición  ${}^{nf}P_0$ .

$${}^{nf}T_0 = \begin{bmatrix} 1 & 0 & 0 & {}^{nf}P_{0x} \\ 0 & 1 & 0 & {}^{nf}P_{0y} \\ 0 & 0 & 1 & {}^{nf}P_{0z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad ; \text{ donde } nf = 4, 5 \text{ y } 6$$

Sólo falta calcular la longitud del actuador, que esta dada por la magnitud del vector y se expresa en forma general como:

$$L_i = \sqrt{({}^{nf}P_{nm x})^2 + ({}^{nf}P_{nm y})^2 + ({}^{nf}P_{nm z})^2}$$

donde  $i = 1, 2, 3, 4, 5, 6$

$nf = 4, 5 \text{ y } 6$ ;  $nm = 1, 2 \text{ y } 3$

Para ejemplificar a continuación se presenta el desarrollo matemático para obtener la longitud del actuador 1.

$${}^6P_2 = \begin{bmatrix} {}^6P_{2x} \\ {}^6P_{2y} \\ {}^6P_{2z} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & {}^6P_{0x} \\ 0 & 1 & 0 & {}^6P_{0y} \\ 0 & 0 & 1 & {}^6P_{0z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi & r_x \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi & r_y \\ -s\theta & c\theta s\psi & c\theta c\psi & r_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} {}^0P_{2x} \\ {}^0P_{2y} \\ {}^0P_{2z} \\ 1 \end{bmatrix}$$

$$L_1 = \sqrt{({}^6P_{2x})^2 + ({}^6P_{2y})^2 + ({}^6P_{2z})^2}$$

### 3.3.3 Matriz de rotación con vector auxiliar

El desarrollo del problema es idéntico al realizado en el caso anterior, sólo que se diferencia en la matriz de rotación, ya que en este caso se genera un vector  $\hat{k}$  ( $k_x, k_y, k_z$ ) el cual, al girar un ángulo  $\alpha$ , permite orientar al nuevo sistema de referencia.

La matriz de rotación con un eje arbitrario es la siguiente:

$$R_{i,\alpha} = \begin{bmatrix} k_x^2 v\alpha + c\alpha & k_x k_y v\alpha - k_z s\alpha & k_x k_z v\alpha + k_y s\alpha \\ k_x k_y v\alpha + k_z s\alpha & k_y^2 v\alpha + c\alpha & k_y k_z v\alpha - k_x s\alpha \\ k_x k_z v\alpha - k_y s\alpha & k_y k_z v\alpha + k_x s\alpha & k_z^2 v\alpha + c\alpha \end{bmatrix}$$

$$\text{donde } v\alpha = \text{vers}\alpha = 1 - \cos\alpha$$

Por lo tanto la matriz de transformación se expresa como:

$${}^0T = \begin{bmatrix} k_x^2 v\alpha + c\alpha & k_x k_y v\alpha - k_z s\alpha & k_x k_z v\alpha + k_y s\alpha & r_x \\ k_x k_y v\alpha + k_z s\alpha & k_y^2 v\alpha + c\alpha & k_y k_z v\alpha - k_x s\alpha & r_y \\ k_x k_z v\alpha - k_y s\alpha & k_y k_z v\alpha + k_x s\alpha & k_z^2 v\alpha + c\alpha & r_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Esta matriz se sustituye en las ecuaciones:

$${}^n P_{nm} = {}^n T_0 {}^0 T {}^0 P_{nm}$$

$$\text{donde } \begin{aligned} n f &= 4, 5 \text{ y } 6 \\ n m &= 1, 2 \text{ y } 3 \end{aligned}$$

De igual manera se obtiene la longitud de cada actuador,

$$L_i = \sqrt{({}^n P_{nm} x)^2 + ({}^n P_{nm} y)^2 + ({}^n P_{nm} z)^2}$$

$$\text{donde } \begin{aligned} i &= 1, 2, 3, 4, 5 \text{ y } 6 \\ n f &= 4, 5 \text{ y } 6 \\ n m &= 1, 2 \text{ y } 3 \end{aligned}$$

### 3.4 Generación de trayectorias

Existe una infinidad de trayectorias que se pueden generar en el espacio con los movimientos del hexápodo que, además, tienden a ser complicadas, pero gracias a la cinemática, que proporciona todos los resultados, la generación de trayectorias se vuelve muy sencilla. El método que se puede utilizar para generar trayectorias depende de las necesidades que se requieran, por ejemplo una posible solución sería representar la longitud de cada uno de los actuadores en función del tiempo, mediante la ecuación de la interpolación:

$$L_i = L_{i_0} + \frac{L_{i_f} - L_{i_0}}{t_f - t_0} (t - t_0)$$

donde  $L_i$  es la longitud del actuador  $i$  en un tiempo  $t$   
 $L_{i_0}$  es la longitud inicial de cada actuador  
 $L_{i_f}$  es la longitud final de cada actuador  
 $t_f$  es el tiempo total para realizar la trayectoria  
 $t_0$  es el tiempo inicial que se puede considerar cero

Esta propuesta permite realizar una trayectoria con la particularidad de que, si se colocan motores de pasos en los actuadores, el resultado sería que todos girarían a diferentes velocidades, porque cada actuador tendrá que recorrer una distancia particular, aunque todos girarían el mismo tiempo.

## 4. DINÁMICA

La dinámica define la energía necesaria que se requiere para variar o mantener la longitud de los actuadores. El objetivo de este capítulo se limita a descomponer la fuerza de gravedad, producida por el peso de la plataforma móvil, en un vértice, para así poder determinar la fuerza que ejerce un actuador como reacción a la fuerza de gravedad (Figura 4.1). El cálculo de la fuerza que ejerce cada actuador permite realizar un criterio de selección, la elección de los actuadores. Esta elección se define determinando la variable fuerza del actuador, el torque en un motor a pasos de un sistema de actuadores mecánicos, o la presión de un compresor para actuadores neumáticos.

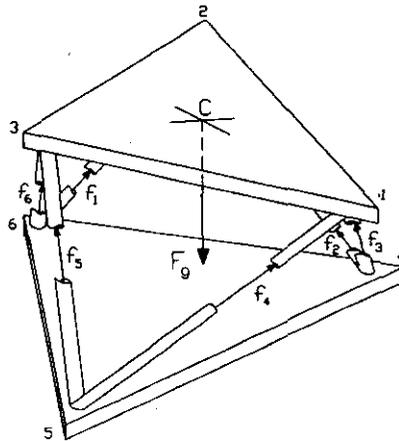


Figura 4.1 Distribución de fuerzas

## 4.1 Desplazamientos Virtuales

El método que se desarrolló para analizar la dinámica es el de los desplazamientos virtuales, basado en la conservación de la energía.

El concepto de este método es analizar la fuerza que requiere un actuador como reacción a la fuerza de gravedad, de la siguiente manera: para una configuración dada se incrementa la longitud de un actuador; esto implica que la plataforma móvil sufre un desplazamiento, de manera que mediante la Cinemática Directa se determinan los nuevos vectores de posición que definen a los vértices de la plataforma; para así obtener la diferencia con respecto a sus valores iniciales, y aplicar posteriormente la ecuación 4.1 de los desplazamientos virtuales.

$$f_a = f_g \frac{\Delta_z}{\Delta_x} \quad (4.1)$$

donde

$f_a$  es la fuerza que ejerce el actuador como reacción a la fuerza de gravedad.

$f_g$  es la fuerza de gravedad, el peso de la plataforma móvil.

$\Delta_z$  es el desplazamiento que sufrió el vector de posición del centro del triángulo en la dirección de la fuerza de gravedad.

$\Delta_x$  es el incremento o decremento que se le aplicó a la longitud del actuador en análisis.

### 4.1.1 Análisis matemático

Para determinar la fuerza  $f_a$  de la ecuación 4.1 es necesario incluir las variables  $f_g$  y  $\Delta_x$ , peso de la plataforma e incremento del actuador respectivamente. Por otro lado, la diferencia que resulta en los vectores de posición de los vértices del triángulo móvil  $\Delta_x$  en dirección de la gravedad se obtiene de la siguiente manera: después de haber variado la longitud de un actuador, se repite el análisis de la cinemática directa para obtener los nuevos vectores de los vértices.

$$\hat{P}_i = (P_{ix}, P_{iy}, P_{iz}) \quad (i=1,2,3)$$

Se obtienen las diferencias con respecto a sus valores iniciales:

$$\Delta \hat{P}_i = (\Delta P_{ix}, \Delta P_{iy}, \Delta P_{iz}) = (P_{ix} - (P_{ix})_0, P_{iy} - (P_{iy})_0, P_{iz} - (P_{iz})_0) \quad (i=1,2,3)$$

El desplazamiento del centro del triángulo estará dado por la media de las componentes de los vértices:

$$\Delta P_c = \left( \sum_{i=1}^3 \frac{\Delta P_{ix}}{3}, \sum_{i=1}^3 \frac{\Delta P_{iy}}{3}, \sum_{i=1}^3 \frac{\Delta P_{iz}}{3} \right) \quad (4.2)$$

Para obtener el incremento del desplazamiento del centro del triángulo en la dirección de la gravedad ( $\Delta_g$ ) hay que definir al vector gravedad ( $v_g$ ). Para cualquier estado físico, el vector gravedad es ortogonal al plano de la Tierra, pero como se está refiriendo cada parámetro con respecto al sistema de referencia de la plataforma fija, es necesario generar un vector que simule la dirección de la gravedad.

$$v_g = (v_{gx}, v_{gy}, v_{gz}) \quad (4.3)$$

Mediante el producto escalar entre los vectores  $\Delta P_c$  y  $v_g$ , ecuaciones 4.2 y 4.3, se obtiene  $\Delta_g$ .

$$\Delta_g = \Delta P_{cx} \cdot v_{gx} + \Delta P_{cy} \cdot v_{gy} + \Delta P_{cz} \cdot v_{gz} \quad (4.4)$$

Finalmente se sustituye el valor obtenido de la ecuación 4.4 en la ecuación 4.1 y se obtiene la fuerza  $f_a$ .

## **5. CONTROL DEL ESPEJO SECUNDARIO DE UN TELESCOPIO**

*Como se expuso en el Objetivo, este capítulo se refiere a una aplicación del hexápodo como el manipulador que controla al espejo secundario de un telescopio. Esta aplicación forma parte de un problema real, por lo que a continuación se dará una introducción de ¿qué es y cómo funciona un telescopio?; una mención esquemática de ¿qué es y cómo se realizan las investigaciones astronómicas?; para que así se pueda cuestionar ¿dónde y por qué se presenta el problema?; y finalmente se describirá la propuesta de cómo resolver dicho problema.*

### **5.1 Introducción**

*Un telescopio cumple una doble función, la de proporcionar la imagen de un objeto que se encuentra a una cierta distancia y además la de recolectar la mayor cantidad de luz proveniente de dicho objeto. Los telescopios astronómicos se dividen en tres tipos: sistemas dióptricos (refractores), sistemas catóptricos (reflectores), y sistemas catadióptricos (objetivos de lentes y espejos). [ 7 ]*

*Las características ópticas fundamentales de los telescopios son: el diámetro de la pupila de entrada y la distancia focal.*

### 5.1.1 Clases de Telescopios

**Los telescopios refractores.**- La óptica más simple de un telescopio refractor está compuesta por un sistema de dos lentes convergentes, el objetivo y el ocular. El objetivo es la lente que recibe la luz del objeto o cuerpo celeste y su función es la de formar una imagen de éste en el plano focal ( $f$ ). Esta imagen puede ser vista mediante un ocular constituido por la segunda lente que formará la imagen sobre la retina del ojo, más grande y con mayor resolución de la que se formaría si se viera sin la ayuda del telescopio.

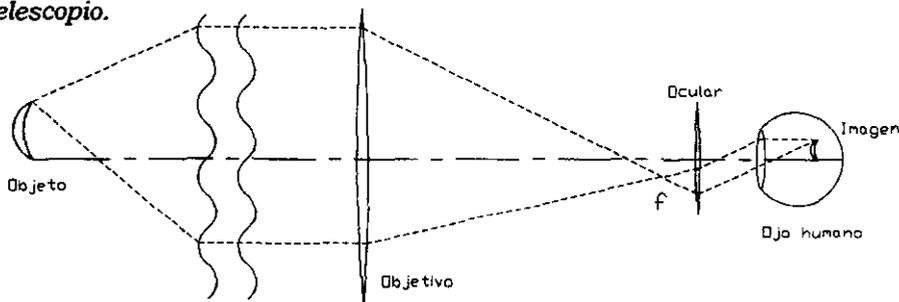


Figura 5.1 Esquema óptico de un sistema refractor

La distancia que hay entre las dos lentes es variable y el usuario definirá la distancia hasta que la imagen alcance el enfoque deseado. Entre más grande sea el diámetro del objetivo, mayor será la cantidad de luz recibida.

Las desventajas de un telescopio refractor son tales que se prefiere en los telescopios modernos el modo de reflexión; estas desventajas son:

- La absorción de algunos colores de la luz cuando atraviesa un medio refractor.
- El ángulo de refracción es una función de la longitud de onda, que genera aberraciones, llamadas **aberraciones cromáticas**. Este fenómeno de la refracción es el mismo que ocurre cuando la luz pasa a través de un prisma y se descompone en colores.
- En cuanto al sistema estructural se detecta otro inconveniente. Como se dijo anteriormente, entre más grande sea el objetivo, mayor será la cantidad de luz que el telescopio recibirá. Para evitar las aberraciones ópticas con el aumento del diámetro del objetivo se tendrá que aumentar su distancia focal. Esto dará como resultado un telescopio sumamente largo, que se verá afectado por flexiones estructurales.

**Telescopios reflectores.**- El sistema óptico que se emplea en un telescopio reflector consta de un espejo primario que concentra la luz en un foco  $f'$  y de un espejo secundario que la dirige hacia un foco  $f$ , donde se proyectará la imagen del cuerpo celeste.

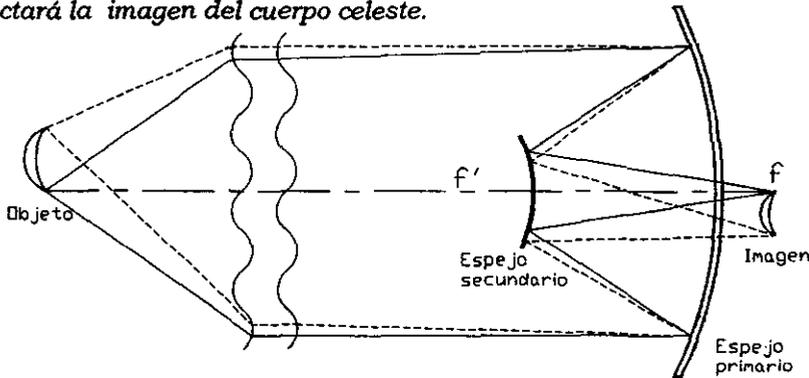
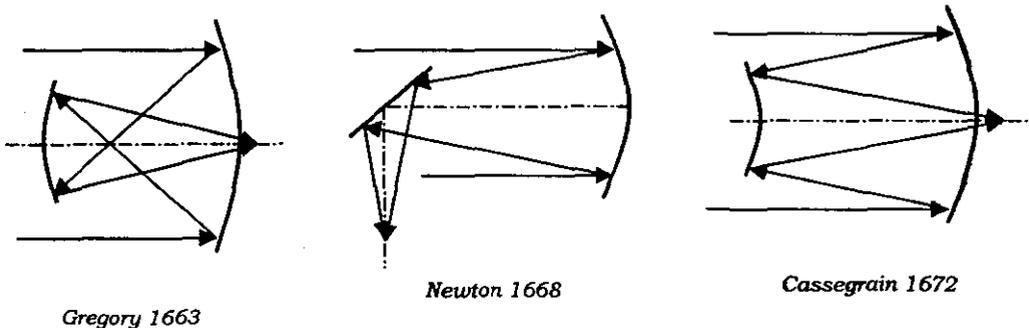


Figura 5.2 Esquema óptico de un sistema reflector.

Las ventajas que presenta el sistema reflector con respecto al refractor son:

- La reflexión de un buen espejo hace que la absorción de la luz sea mínima.
- El ángulo de reflexión es el mismo para todos los colores.
- El diseño estructural se reduce en cuanto a la longitud, ya que a paridad de resolución la distancia focal del espejo primario puede ser más pequeña que la distancia focal del objetivo refractor.

Dependiendo de la configuración óptica, existen varios tipos de telescopios reflectores; a continuación se muestran algunos de los esquemas más comunes, con el nombre de su inventor y el año de su descubrimiento.



Gregory 1663

Newton 1668

Cassegrain 1672

Figura 5.3 Los esquemas ópticos más comunes de los telescopios reflectores

Los esquemas a lo largo del tiempo fueron evolucionando, pero conceptualmente el modelo de Cassegrain es el que se utiliza en la mayoría de los observatorios astronómicos, mientras que el modelo de Newton es el utilizado en los pequeños telescopios personales.

Entre los modelos que aplican el principio de Cassegrain se encuentra el de Coudé. Este modelo sólo cambia en la posición del foco, ya que la luz reflejada por el espejo secundario está desviada por un conjunto de espejos a un lugar fijo del edificio, (ver figura 5.4) donde se pueden colocar instrumentos pesados, como espectrógrafos. Otro ejemplo sería el de Ritchey-Chretien, que es una versión más avanzada del modelo de Cassegrain, sólo que las superficies son más complicadas y permiten una mejor calidad de imagen. El foco de Nasmyth (ver figura 5.4) es una opción para colocar instrumentos detectores pesados a un lado del sistema óptico.

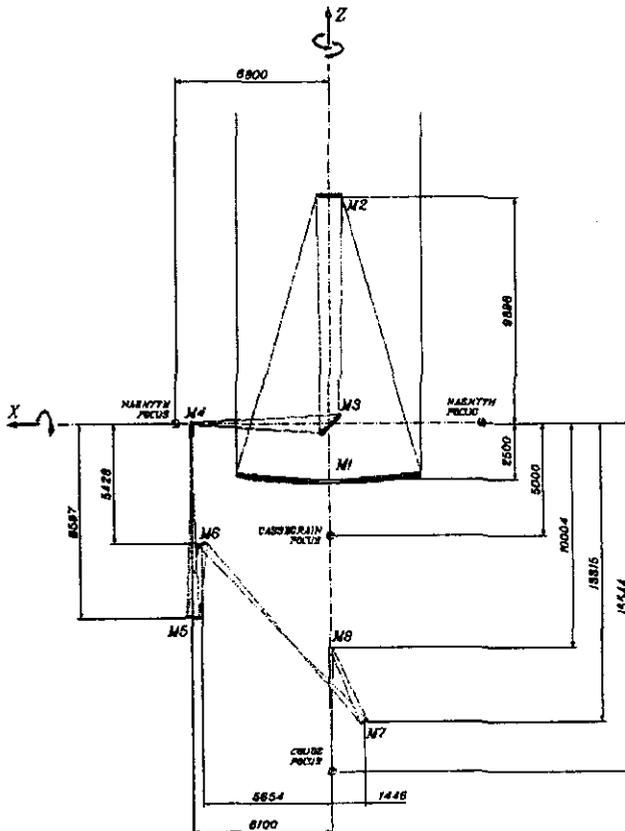


Figura 5.4 Sistema óptico del Telescopio ESO

**Telescopios catadióptricos.**- El sistema catadióptrico se forma gracias a la combinación de superficies reflectoras (espejos) y elementos refractores (lentes). En este caso, el papel fundamental para formar la imagen lo tiene el espejo o en su caso un par de ellos, mientras que los lentes tienen la función de reducir las aberraciones.

El ejemplo más claro de este tipo de sistema óptico lo presenta el telescopio o cámara de Schmidt (1932). Consiste en un espejo esférico cuya aberración es eliminada por medio de una lámina correctora, una de cuyas caras está tallada según una superficie de perfil calculada para que las aberraciones por ella producidas anulen las producidas por el espejo. La cámara de Schmidt es muy utilizada para fotografías de la bóveda celeste, debido a su gran ángulo de visión. [ 8 ]

### Telescopio de Cassegrain y análogos

El telescopio de Cassegrain, como se mencionó anteriormente, es el modelo que se utiliza en los observatorios astronómicos, como se muestra en las figuras 5.4 y 5.5. Este modelo está formado por un espejo primario, con superficie parabólica, que recibe la luz del cuerpo celeste y la refleja a un segundo espejo o espejo secundario, el cual es hiperbólico. Ambos espejos se encuentran sobre el mismo eje óptico, por lo que el haz reflejado regresa nuevamente hacia el espejo primario y, pasando a través de un orificio de éste, forma la imagen en un plano focal (foco de Cassegrain) en la parte posterior del telescopio, donde se monta un instrumento detector de la luz.

En la figura 5.5 se puede observar un corte del telescopio de mayor diámetro al momento (12m), el telescopio Keck. La luz que llega se refleja en el espejo primario, formado por 36 segmentos; posteriormente la luz se dirige al espejo secundario, que la refleja hacia el orificio del espejo primario donde produce el foco Cassegrain. Se puede incluir un tercer espejo que refleja el haz de luz para crear un segundo foco, el foco de Nasmyth.

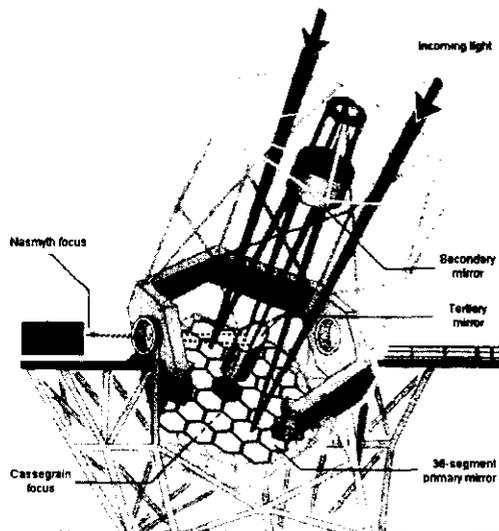


Figura 5.5 Telescopio Keck

### 5.1.2 Movimientos de un telescopio

Durante una observación astronómica un telescopio tiene que estar en continuo movimiento para compensar la rotación de la Tierra y para apuntar las regiones deseadas del cielo. La Tierra, como es sabido, tiene dos movimientos, el de rotación y el de translación. La translación, es el movimiento cuya trayectoria describe una órbita elíptica alrededor del Sol, y su duración es la que determina un año; para un astrónomo la translación implica el cambio de escenario dentro de la bóveda celeste. La rotación de la Tierra es el movimiento que determina la duración de un día, y es el giro alrededor del eje terrestre. El eje terrestre por efectos gravitatorios del Sol y la Luna oscila en un movimiento de tipo giroscópico, llamado de precesión. Cuando se observa la bóveda celeste, la rotación de la Tierra da lugar a que parezca que el cielo esté girando alrededor de ella, y si se desea estudiar un cuerpo celeste, el telescopio tiene que "seguir" este movimiento, esta operación del telescopio se llama movimiento de guiado. Por lo tanto se puede decir que el mecanismo que determina el movimiento de guiado de un telescopio representa un reloj de muy alta precisión.

Para apuntar un telescopio hacia una región específica de la bóveda celeste es necesario realizar dos movimientos fundamentales, el de la ascensión recta y el de la declinación, que corresponden respectivamente a la longitud y a la latitud de la Tierra, solo que referidos a la bóveda celeste.

La ascensión recta se mide hacia el este a lo largo del ecuador celeste. El punto de partida se da cuando el Sol cruza el ecuador de sur a norte, en el equinoccio de la primavera, ese punto se llama Primer Punto de Aries ( $\gamma$ ). Las unidades que se utilizan en la ascensión recta son de tiempo, horas, minutos, segundos. Por lo tanto el Primer Punto de Aries corresponde a  $0^h$ . La presencia de estas unidades se debe porque efectivamente la ascensión recta representa un movimiento análogo a un reloj.

La declinación se mide en grados, minutos y segundos de arco, al norte (+) o al sur (-) del ecuador celeste. Por consiguiente, las coordenadas celestes varían entre  $0^h$  y  $24^h$  ( $=0^h$ ) en cuanto a la ascensión recta, mientras que para la declinación varían de  $+90^\circ$  a  $-90^\circ$ . La importancia de estos movimientos para los astrónomos se debe a que ellos utilizan las coordenadas para determinar la posición de un cuerpo en la bóveda celeste.

[ 9 ]

### 5.1.3 Monturas del telescopio

*El tubo del telescopio que incluye el espejo primario y secundario, tiene que poseer los movimientos para orientarse hacia cualquier punto de la bóveda celeste y para "seguir" la rotación de la misma. Para facilitar estos movimientos, garantizar un mínimo de flexiones mecánicas y optimizar los costos, se han venido desarrollando varios diseños mecánicos de monturas que han aprovechado los recursos tecnológicos de la época.*

*La montura más convencional basada en recursos esencialmente mecánicos es la montura ecuatorial. En este caso el soporte principal está constituido por una horqueta o un yugo capaces de rotar alrededor de un eje paralelo al eje de rotación de la Tierra y, sobre este soporte el tubo del telescopio está montado mediante una articulación de rodamientos que permite una rotación sobre un eje ortogonal al eje de rotación de la Tierra. El primer movimiento se llama de ascensión recta, el segundo movimiento de declinación. Mediante la combinación de los dos movimientos el telescopio puede ser orientado hacia cualquier objeto de la bóveda, mientras que el sólo movimiento de ascensión recta proporciona el movimiento de guiado. Otra virtud de esta montura consiste en el hecho que la orientación de la imagen en el plano focal Cassegrain queda invariada durante el movimiento de guiado.*

*Una montura alternativa que en tiempos más recientes se ha vuelto muy competitiva es la montura altazimutal. En este caso el soporte principal está constituido por una horqueta cuyo eje de rotación es vertical, mientras que la articulación del tubo del telescopio sobre la horqueta permite un movimiento de rotación alrededor de un eje horizontal. El primer movimiento se llama de acimut, el segundo de altitud. También en este caso ambos movimientos permiten la orientación del telescopio hacia cualquier punto de la bóveda celeste, mientras que para la operación de guiado se tienen que utilizar ambos movimientos de manera sincronizada. Esta sincronización se logra mediante un control basado en el uso de computadoras. También mediante el mismo control es necesario seguir el cambio de orientación presentado por una imagen en el plano focal, durante la operación de guiado. La complejidad de la operación de control en una montura altazimutal se ve altamente recompensada por la rigidez de la estructura mecánica y por el costo de la misma. Otra ventaja es que el uso del foco Nasmyth resulta más adecuado. [ 10 ]*

## **5.2 Descripción del problema**

Los problemas que se presentan en la actualidad en los telescopios que no contienen un manipulador que controle el espejo secundario son relevantes. Un ejemplo es el telescopio de dos metros de diámetro del Observatorio de San Pedro Mártir, en Baja California. A continuación se presentan las necesidades de un astrónomo y cuáles son los problemas que genera la falta de un control en el espejo secundario del telescopio.

Como se explicó anteriormente la necesidad en un sistema de Cassegrain, es que ambos espejos se encuentren con el eje óptico colineal. Esto implica que cualquier desalineamiento provoca aberraciones en la imagen. La causa principal se debe a las deformaciones que puede presentar la estructura mecánica que soporta al sistema óptico del telescopio. Este problema se incrementa entre más inclinado esté el telescopio. En el Observatorio de San Pedro Mártir este problema se resuelve de forma "manual", ajustando mediante tornillos la posición del espejo secundario; además de la pérdida de tiempo, este ajuste se realiza sólo con el telescopio en forma vertical, donde no se consideran las flexiones cuando el telescopio está inclinado. Una solución parcial consistiría en compensar las flexiones mediante la orientación del telescopio, pero esto resulta ser engañoso, ya que este método produce desviaciones en las coordenadas celestes y corrige solo en parte las aberraciones ópticas.

La idea que aquí se propone permite desarrollar los siguientes puntos:

1. Controlar el telescopio en forma automática.
2. Alinearlo para cualquier configuración.
3. Aumentar la eficiencia de la operación de alineación.
4. Proporcionar precisión a los indicadores de ascensión recta y declinación.
5. Disponer de un aparato metrológico.

El control, en forma automática, permite que el sistema no sea manipulado por un operador; esto representa una gran ventaja si se desea controlar al telescopio de forma remota, por ejemplo un telescopio espacial. En un telescopio espacial las flexiones no representan un verdadero problema ya que se encontraría en un estado sin gravedad, pero tendría importancia para corregir y optimizar la imagen.

La capacidad de alinearse para cualquier configuración representa una extraordinaria ventaja sobre las alineaciones estáticas, ya que estas últimas, como se explicó anteriormente en el caso del Observatorio de San Pedro Mártir, se realizan solamente en el cenit, o sea con el telescopio en forma vertical, mientras que una alineación automática puede compensar la

*desalineación mecánica para cualquier orientación del telescopio distinta a la posición cenital.*

*El proceso es sumamente rápido si se compara con el método manual. En el modo manual se tienen muchos errores humanos; además de que se necesita un equipo óptico externo que permita realizar las pruebas de error para poder alinear a los espejos, y resulta un factor contraproducente puesto que al separarlo del sistema se alteraría nuevamente la alineación. Esta propuesta tiene la capacidad de alinear a los dos espejos con la ventaja, como se verá más adelante, de tener el "test" óptico integrado.*

*Cuando el telescopio está inclinado y el objeto que tendría que estar en el centro del ocular o del instrumento de investigación se sale de la visión, debido a las deformaciones estructurales, el astrónomo mueve la imagen del objeto que está observando cambiando la posición del telescopio, es decir poniendo el telescopio en una posición "falsa". Lo mejor es siempre que el telescopio tenga en el centro del campo la región del cielo con las coordenadas (ascensión recta y declinación) nominales.*

*Otra función que puede alcanzar el hexápodo es la de ser un instrumento metrológico. La medición que se puede realizar con el hexápodo es la que permite determinar la deformación que sufrió la estructura. Por ejemplo para una inclinación dada que tiene el telescopio (declinación y ascensión recta), se puede calcular cual sería la posición que debería tener el espejo secundario sin considerar las deformaciones que este pueda tener, se hace la diferencia con la posición real y se obtiene la deformación que tuvo la estructura.*

### 5.3 Descripción de la propuesta

La propuesta que aquí se plantea parte del siguiente problema: debido a las flexiones de la estructura del telescopio se crea una desviación (desalineamiento) del secundario con respecto al plano focal, por lo que se necesita un control capaz de determinar la posición real del espejo con respecto al mismo plano focal y corregir dicha desviación.

Los componentes que se incluyeron en esta propuesta son los siguientes: una cámara de vídeo CCD instalado en un foco, preferiblemente el de Nasmyth, (se considera el foco de Nasmyth para que el astrónomo pueda utilizar el foco de Cassegrain para colocar el ocular); tres pequeños leds de diferentes colores colocados en el borde del espejo secundario, de manera que cada led esté situado en cada vértice de la plataforma triangular del hexápodo (Figura 5.7).

La propuesta requiere que la cámara televisiva (CCD) monitoree las imágenes de los tres leds en una pantalla televisiva o monitor, para que posteriormente con un software se interprete la posición de las imágenes. Se define, como se verá más adelante, la posición final de los puntos imágenes (puntos de referencia) con el propósito que los puntos imágenes "reales" coincidan con los "de referencia"; para esto se manipula la longitud de cada actuador del hexápodo, dando como resultado que la posición de las imágenes de los tres leds varíe. Se aplica un programa de control que genera ciclos que permiten variar las imágenes hasta alcanzar la coincidencia entre los puntos "reales" y "de referencia". En esta condición el espejo secundario quedaría alineado con respecto al plano focal.

Este sistema representa una prueba óptica muy simple, que puede ser la base para una prueba óptica más compleja. Tiene la propiedad de evaluar en **tiempo real** el estado del hexápodo. Esta operación se puede describir de la siguiente forma: las imágenes de los leds representan los vértices de la plataforma móvil del hexápodo, que es un triángulo equilátero, pero si se observa la plataforma desde un ángulo inclinado, la perspectiva generada por las imágenes de los leds formarán un triángulo escaleno. Esta deformación del triángulo imagen con respecto a la posición final que deberían alcanzar los puntos (un triángulo equilátero), representa el criterio de comparación que aplicará el control.

El plano imagen de la cámara televisiva, que a lo largo del texto se llamará simplemente plano imagen, es el plano donde se proyectan las imágenes de los leds a través de sus lentes internas, misma imagen que se observa en el monitor. El plano imagen se encuentra perpendicular al eje óptico del espejo primario, además como ya se explicó en un modelo de Cassegrain, el eje óptico del espejo secundario debe ser colineal al del

espejo primario. Esto implica que cuando los espejos están alineados, el plano imagen será perpendicular al eje óptico del espejo secundario, dando como resultado que los puntos imágenes formen un triángulo equilátero, pues la perspectiva de un triángulo equilátero visto ortogonalmente mantiene su forma, y sólo cambiará el tamaño en función de la distancia que hay entre los leds y la cámara.

### 5.3.1 Los seis grados de libertad del espejo

Los seis grados de libertad del espejo están distribuidos de la siguiente manera:

De los seis grados de libertad del espejo, uno no interviene en el problema: la rotación del espejo alrededor de su mismo eje. De acuerdo al sistema de referencia  $O^*$  de la figura 5.6, esta rotación representa el giro. Un segundo grado de libertad, el enfoque, es una variable definida por el astrónomo y representa el movimiento longitudinal a lo largo del eje óptico, el eje  $Z^*$  de la misma figura.

Los otros cuatro grados de libertad son fundamentales para alinear al espejo secundario. Dos permiten posicionar al espejo en el plano perpendicular al eje óptico (la posición con respecto al plano generado por los ejes  $X^*$  y  $Y^*$ ). Los últimos dos grados de libertad son los ángulos que representan la orientación del espejo (desviación y elevación).

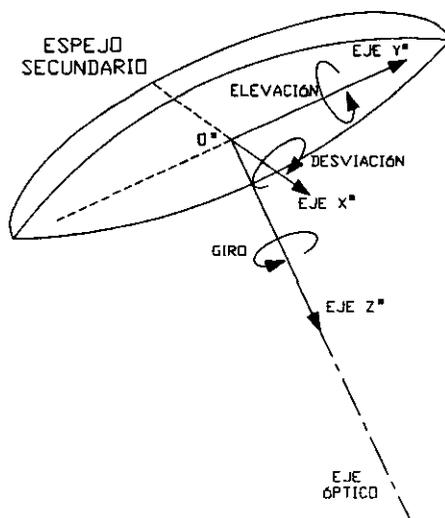


Figura 5.6 Rotaciones del espejo secundario

### 5.3.2 Sistema Real (Automatización)

La automatización consiste en un programa para determinar el orden de los eventos, así como para instruir al sistema sobre como debe llevarse a cabo cada uno de los pasos de la operación. "Autómata": máquina que imita la figura y movimientos de un ser animado. Mientras que la definición de la Enciclopedia Británica dice que "La automatización es el desempeño de operaciones automáticas dirigidas por medio de comandos programados con una medición automática de la acción, retroalimentación y toma de decisiones" [ 11 ].

Estas definiciones comparadas con lo mencionado en la "Descripción de la Propuesta (5.3)" permiten expresar a la aplicación del hexápodo en un telescopio como un sistema automático.

Para poder aplicar el control automático es necesario integrar el manipulador con los tres leds, el sistema de lectura de tiempo real mediante una cámara televisiva CCD y un software capaz de interpretar las imágenes generadas. Integrado el sistema se emplea un programa que ejerza el control. Por ser una aplicación que requiere de una alta precisión sus costos serán elevados. Para su comprensión a continuación se presenta un esquema del sistema:

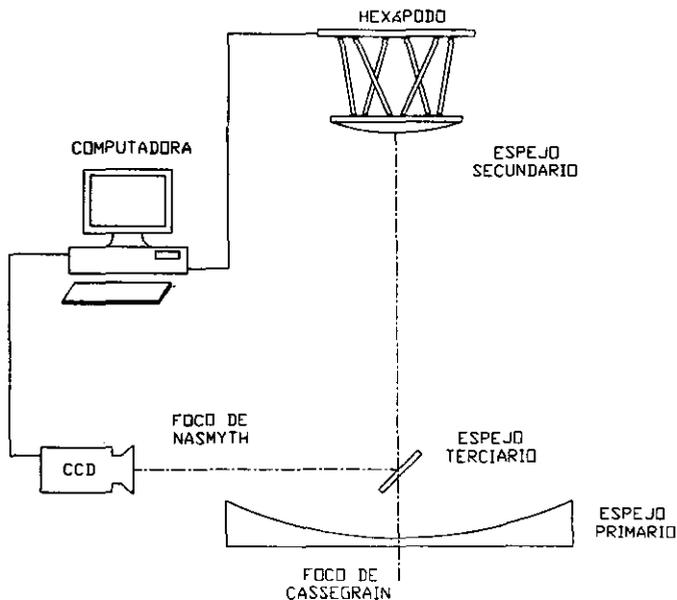


Figura 5.7 Esquema del sistema.

## 5.4 Simulación.

Dado un sistema real, una simulación de éste es un modelo equivalente en cuanto a funcionamiento y construido mediante elementos que, en la mayoría de los casos, son derivados de las matemáticas, (vectores, ecuaciones diferenciales, etc.). La idea de realizar una simulación es una alternativa que permite comprobar la eficacia y eficiencia del programa de control, sin la necesidad de invertir económicamente para construir el sistema real. El programa de control es el mismo tanto para la simulación como para el sistema real, de igual forma las variables de entrada que requiere la teoría del control serán las mismas en ambos casos. En el problema que se está tratando estas variables son las proyecciones de los leds en el plano imagen.

Para generar la simulación hay que definir las condiciones que se presentan en el sistema real, para de esta forma poder desarrollar las variables de entrada. Estas condiciones son:

- La posición actual del espejo secundario con respecto al plano imagen de la cámara.
- La distancia deseada entre el espejo secundario y el plano imagen de la cámara.
- La posición del centro de la lente de la cámara.

El primer punto en el sistema real viene dado por la prueba óptica basada en los tres leds, que al ser proyectados en el plano imagen permiten evaluar la posición del hexápodo. En la simulación se logra proyectar las imágenes por medio de rectas que simulan los rayos emitidos por los leds, para generar estas rectas es necesario definir la posición del hexápodo con respecto al plano imagen de la cámara y esto se logra incluyendo una matriz de transformación

El enfoque es el movimiento que realiza el astrónomo para afinar o ajustar la imagen de un cuerpo celeste. Un ejemplo muy sencillo es el de la cámara fotográfica, donde el usuario para enfocar un objeto varía la distancia de los lentes. En el caso del telescopio la distancia que permite enfocar a los cuerpos celestes es la que existe entre el espejo secundario y el plano focal donde se encuentra el instrumento de medición y, una vez alcanzado el enfoque, tendrá que mantenerse invariable a lo largo de las observaciones astronómicas. La distancia está dada a partir del diseño óptico del telescopio. Para desarrollar la simulación hay que incluir el valor de esta distancia.

Considerando que la cámara tiene una lente delgada, la posición del centro de esta lente define una distancia con respecto al plano imagen de la cámara. La variación de esta distancia altera el tamaño de las imágenes de la siguiente manera: entre más cerca esté la lente al plano imagen, más pequeña será la imagen; de forma inversa, entre más grande es la distancia, más grande será la imagen que proyecta. En la simulación se debe incluir el valor de esta distancia.

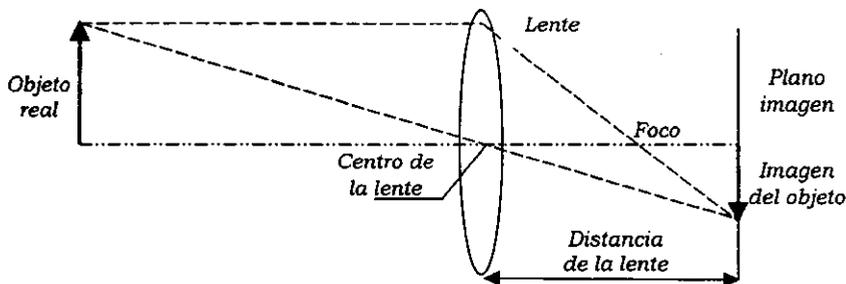


Figura 5.8 Esquema del sistema óptico de la cámara

Nota: Si se considera para la cámara televisiva un objetivo con varias lentes (multiplete), el centro de la lente estará sustituido por los "puntos principales", de manera que la distancia entre el led y el centro de la lente, así como la distancia entre la imagen y el centro de la lente serán sustituidos por la distancia entre el led y el primer punto principal y la distancia entre la imagen y el segundo punto principal, respectivamente.

### 5.4.1 Referencia

Ya se mencionó que para desarrollar la simulación hay que definir la posición actual del espejo con respecto al plano imagen de la cámara. Esto implica que en el plano imagen de la cámara sea necesario generar un nuevo sistema de referencia ( $O'$ ). Los ejes de este sistema estarán dispuestos de la siguiente manera: el plano formado por los ejes  $X'$  y  $Y'$  es precisamente el plano imagen, mientras que el eje  $Z'$  toma el papel de eje óptico.

Como la posición del espejo secundario es equivalente a la posición del triángulo móvil del hexápodo, la referencia del espejo secundario se obtendrá mediante una matriz de transformación que relacionará al triángulo móvil y al nuevo sistema de referencia que define al plano imagen.

La matriz de transformación 5.1 que se emplea, se generó con la matriz de rotación de Euler y el vector de posición que describe el centro del triángulo móvil con respecto al sistema de referencia  $O'$ .

En el momento de correr la simulación, el usuario deberá estimar los seis valores  $(Pc'_x, Pc'_y, Pc'_z, \psi', \theta', \phi')$ , y la matriz estará generada.

$${}_{o'}T_3 = \begin{bmatrix} c\phi'c\theta' & c\phi's\theta's\psi' - s\phi'c\psi' & c\phi's\theta'c\psi' + s\phi's\psi' & Pc'_x \\ s\phi'c\theta' & s\phi's\theta's\psi' + c\phi'c\psi' & s\phi's\theta'c\psi' - c\phi's\psi' & Pc'_y \\ -s\theta' & c\theta's\psi' & c\theta'c\psi' & Pc'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.1)$$

La matriz de transformación fue introducida para generar las rectas que simulan a los rayos de los leds. Pero la información matemática que proporciona bastaría para que con sólo utilizar geometría analítica se pueda describir la trayectoria que debería seguir el hexápodo para alinear a los dos espejos. Así que en el momento de desarrollar el control no se ocupará la información de dicha matriz. Si se compara este hecho con el sistema real se puede apreciar que los rayos de los leds existen e inciden en el plano imagen de la cámara televisiva, pero se desconoce la posición que tiene el hexápodo con respecto a la cámara.

Los sistemas de referencia que se han empleado se relacionan mediante matrices de transformación, como se ve en la figura 5.9:

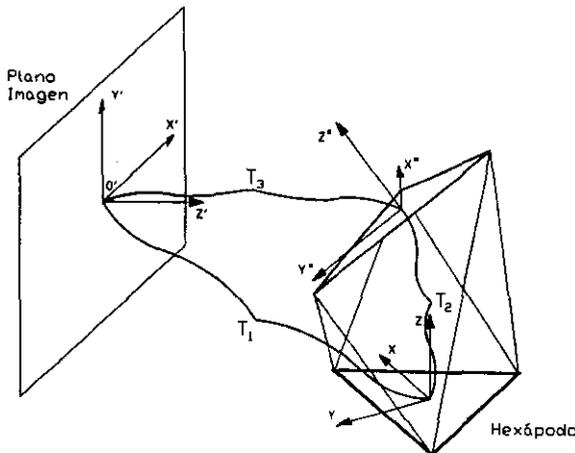


Figura 5.9 Configuración de las matrices de transformación

La función de cada matriz se representa en la siguiente tabla.

$T_1$	Define a la estructura del telescopio.
$T_2$	Define a la matriz de transformación del hexápodo.
$T_3$	Permite expresar las proyecciones de los leds.

### 5.4.2 Distancia del lente de la cámara

Para poder generar los puntos imagen es necesario definir cual es la distancia del lente de la cámara con respecto al plano imagen de la misma. El sistema de referencia  $O'$  fue construido de tal manera que el eje  $Z'$  sea el eje óptico, lo cual implica que el centro de la lente ( $Clente$ ) se encuentre en un punto sobre este eje. Como se mencionó anteriormente, esta distancia es un valor que se proporciona para generar la simulación.

Por razones de imagen en el monitor de la computadora es conveniente que la distancia del centro de la lente de la cámara al plano imagen de la misma ( $Dlente$ ) sea:

$$Dlente = 2LM \left[ 1 - \left( \frac{LM}{LM + LMi} \right) \right] \quad (5.2)$$

**LM** es la longitud de cada lado del triángulo móvil.

**LMi** es la longitud de cada lado del triángulo imagen.

Esta fórmula ha sido obtenida de la siguiente manera: entre las variables de entrada que se deben incluir para generar la simulación está la distancia deseada entre el espejo secundario y el plano imagen de la cámara. Se impuso que esta distancia fuese dos veces la longitud del lado del triángulo móvil ( $2LM$ ). De igual forma es necesario definir el tamaño del triángulo equilátero, que tendrán que formar los puntos imágenes, en la pantalla; en este caso la longitud de cada lado  $LMi$  es de 192 pixeles.

La razón de haber elegido 192 pixeles se debe a que el programa de control está en lenguaje Basic y entre los modos de pantalla que dispone está SCREEN 12, que permite tener una resolución de 640 X 480 pixeles. Debido a que 96 pixeles corresponden a una pulgada, cada lado del triángulo imagen será de 2 pulgadas, por lo que la imagen quedará a escala del monitor.

Con estas consideraciones, y en el caso en que el espejo esté alineado, se alcanza la siguiente figura (5.10):

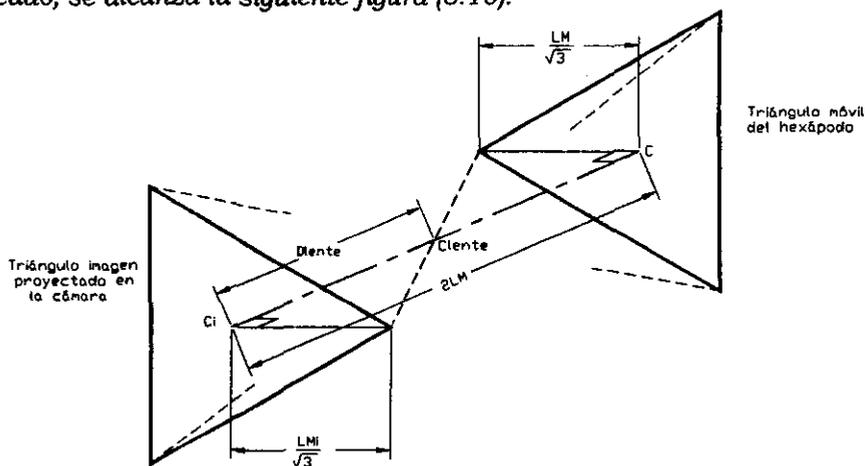


Figura 5.10 Distancia del lente de la cámara

Se puede considerar que la distancia  $D_{lente}$  es una fracción de la distancia deseada entre el espejo secundario y el plano imagen ( $2LM$ ). Esto da como resultado la siguiente expresión, donde  $\alpha$  es el factor que define la fracción:

$$D_{lente} = (1 - \alpha) \cdot 2LM \quad (5.3)$$

Por otro lado de la figura 5.10 se pueden apreciar dos triángulos que tienen como vértice común el centro de la lente. Por triángulos semejantes se tiene:

$$\frac{\frac{LM}{\sqrt{3}}}{(\alpha) \cdot 2LM} = \frac{\frac{LMi}{\sqrt{3}}}{(1 - \alpha) \cdot 2LM} \quad (5.4)$$

resolviendo y despejando  $\alpha$

$$\alpha = \frac{LM}{LM + LMi} \quad (5.5)$$

sustituyendo la ecuación (5.5) en la ecuación (5.3) se comprueba la ecuación (5.2)

$$D_{lente} = 2LM \left[ 1 - \left( \frac{LM}{LM + LMi} \right) \right]$$

### 5.4.3 Proyección de los haces

Definida la posición del centro de la lente se puede deducir donde se encuentran las imágenes de los leds. Para la simulación sólo es necesario crear tres rectas que partan de los vértices del triángulo móvil, crucen el centro de la lente, y posteriormente intersecten el plano imagen. Las tres intersecciones representarán la posición de los puntos imágenes. Para generar las rectas se emplea la ecuación vectorial de la recta:

$$IM_i = \hat{P}_o + t_i \hat{u}_i \quad (5.6)$$

donde  $IM_i$  es el vector de posición del punto imagen.

$\hat{P}_o$  es el vector de posición del centro de la lente.

$t_i$  es el parámetro, un escalar.

$i$  se refiere al led que genera la recta (1, 2 ó 3).

$\hat{u}_i$  es el vector que define la dirección de la recta.

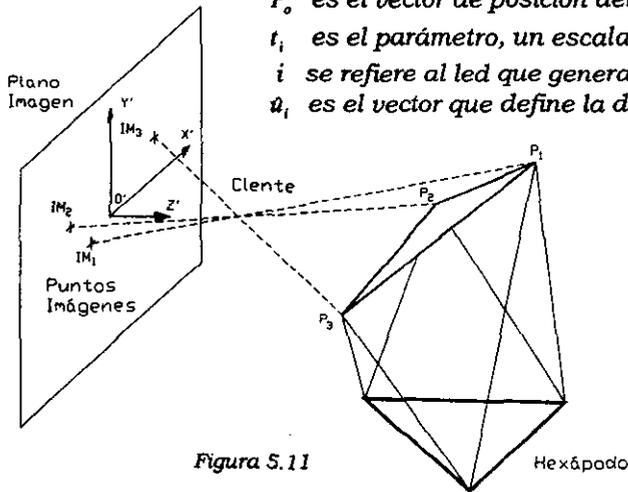


Figura S.11

Al conocer la posición del centro de la lente y de los vértices ( $P_i$ ), se puede determinar cada vector  $\hat{u}_i$

$$\hat{u}_i = \hat{P}_i - \hat{P}_o$$

donde  $\hat{P}_o$  es el vector de posición del centro de la lente:

$$\hat{P}_o = (0, 0, D_{lente})$$

Por otro lado sabemos que:

$$\hat{u}_i = (u_x, u_y, u_z)_i$$

donde:

$$u_x = (P_i)_x \quad u_y = (P_i)_y \quad u_z = (P_i)_z - D_{lente}$$

Proyectar las imágenes de los leds sobre el plano imagen, representado por el plano X'Y' del sistema de referencia, implica que la componente en "z" del vector  $\hat{u}_i$ , tenga un valor nulo. Entonces en la ecuación vectorial de la recta se puede deducir la siguiente expresión:

$$0 = (P_o)_z + t_i (u_z)$$

Otra forma de escribirla sería:

$$0 = D_{lente} + t_i ((P_i)_z - D_{lente})$$

Esto da como resultado que la única variable desconocida es el parámetro  $t_i$ , que al despejarlo:

$$t_i = \frac{D_{lente}}{D_{lente} - (P_i)_z} \quad (5.7)$$

De esta forma se obtiene el valor del parámetro correspondiente al punto imagen para cada una de las ecuaciones de las tres rectas. Ahora solo queda sustituir en la ecuación vectorial o paramétrica de la recta

$$IM_{i(x,y)} = 0 + t_i \hat{u}_{(x,y)}$$

Por lo que los puntos imágenes quedarían:

$$IM_{i_x} = \left( \frac{D_{lente}}{D_{lente} - (P_i)_z} \right) P_{i_x} \quad (5.8 a)$$

$$IM_{i_y} = \left( \frac{D_{lente}}{D_{lente} - (P_i)_z} \right) P_{i_y} \quad (5.8 b)$$

(i = 1, 2, 3)

De esta forma se han generado las imágenes de los leds en el monitor, que permiten obtener la perspectiva en el caso real. En este momento en el caso real faltaría incluir el programa que interprete la posición de los puntos sobre la pantalla, ya que en la simulación, por ser un desarrollo matemático, las coordenadas se obtienen directamente como resultado.

## 5.5 Control Algorítmico.

Lograr que, mediante las imágenes generadas por los tres leds en un plano se puede representar seis grados de libertad, implica un verdadero reto; por esta razón es necesario hacer un sistema de control que permita desarrollar el problema.

### 5.5.1 Definiciones

**Teoría de control moderna.** Controlar significa medir el valor de la variable controlada del sistema y aplicar la variable manipulada al sistema para corregir o limitar una desviación del valor medido a partir de un valor deseado. La variable controlada es la cantidad o condición que se mide y controla. La variable manipulada es la cantidad o condición que el controlador modifica para afectar el valor de la variable controlada. Por otro lado se considera control moderno porque se aplica a sistemas con entradas y salidas múltiples, que pueden ser lineales o no lineales. [ 12 ]

Con estas definiciones se puede comprender el control que se aplica en un hexápodo.

- Las variables controladas son las proyecciones de los leds o puntos imágenes, los cuales se miden con respecto a los puntos de referencia. Por lo general la variable controlada es el resultado del sistema, en este caso la alineación de los dos espejos.
- Las variables manipuladas son las longitudes de los actuadores. Son las variables que se modifican para afectar a las variables controladas, o sea al modificar las longitudes de los actuadores, cambia la posición y orientación de la plataforma móvil, afectando así la posición de los puntos imágenes o proyecciones de los leds.

**Algoritmos.** Un algoritmo es el medio por el que explicamos cómo puede resolverse un problema, utilizando necesariamente una aproximación paso a paso [ 13 ]. Entre los ejemplos que describen a un algoritmo se pueden mencionar a un instructivo o manual de ensamble, a una receta de cocina, hasta a una estructura matemática muy compleja. El algoritmo se caracteriza por ser preciso, puesto que debe indicar el orden de realización de cada paso; si este orden no se sigue al pie de la letra el resultado final puede presentar irregularidades. Un algoritmo debe estar definido, es decir que al repetir el algoritmo se debe obtener el mismo resultado. Finalmente el algoritmo debe ser finito, o sea termina en un número determinado de pasos. Se puede concluir que un algoritmo es un método para resolver un problema mediante una serie de pasos precisos, definidos y finitos.

### 5.5.2 Propuestas de control

Para resolver el problema de la alineación de los espejos de un telescopio, existen muchas formas de aplicar el control. En este proyecto se contemplan tres propuestas estructuradas sobre la base de tres algoritmos: algoritmo raíz, algoritmo primario y algoritmos secundarios, que serán definidos en detalle en cada una de las propuestas.

El algoritmo raíz tiene la finalidad de alinear a los espejos y contiene a su vez a un algoritmo primario y algoritmos secundarios. El algoritmo primario permite, por medio de expresiones matemáticas, aproximar la alineación de los espejos; mientras que los algoritmos secundarios ayudan al primario a alcanzar este propósito, ya sea impidiendo una divergencia o en su caso acelerando el proceso. El algoritmo raíz es **preciso** porque presenta un orden para realizar cada paso, que consiste en ejecutar tanto el algoritmo primario como los secundarios. Está **definido**, ya que si se repite la misma operación se obtendrán los mismos resultados. Además es un desarrollo **finito**, pues a pesar de que el algoritmo está inmerso en un ciclo, éste va a terminar en el momento que el control satisface la condición de salida. La condición de salida es precisamente el estado final que desea alcanzar el algoritmo y representa el parámetro que diferencia a las tres propuestas. Las tres se basan en tener como condición de salida a una imagen de referencia que servirá para hacer las comparaciones entre el estado deseado y el estado actual del sistema, por lo que la condición de salida se considerará como la condición de comparación. Debido a que en cada propuesta la imagen es diferente, se llamará a cada una ellas según la imagen que presenta:

- Triángulo de referencia
- Circunferencia de referencia
- Triángulo de referencia con prueba láser

Recordemos que cuando se describió la propuesta del problema se mencionó que existían “puntos de referencia”, pues bien son éstos los que nos permitirán crear la condición de comparación y salida. La idea en las tres propuestas de control es la misma: una vez determinado el estado del hexápodo con los “puntos imagen”, se proyectan los “puntos de referencia”, o sea la configuración final que deben alcanzar los “puntos imagen” para lograr la alineación de los dos espejos. Posteriormente se realizarán los movimientos del hexápodo para alcanzar la configuración final.

Los movimientos del hexápodo están generados en cada propuesta por el algoritmo primario y por los secundarios y en cada caso, como se verá más adelante, son diferentes. A continuación se presentarán las diferentes propuestas de control.

## 5.6 Triángulo de referencia.

Esta propuesta presenta a los "puntos de referencia" como vértices de un triángulo equilátero, o sea el monitor proyecta este triángulo con el tamaño de los lados de 192 pixeles. El centro de dicho triángulo se encuentra en el origen del sistema de referencia  $O'$ . Los puntos que forman este triángulo son de diferentes colores: azul, verde y rojo (PC); de igual forma los tres leds corresponden a la misma configuración de colores. Para diferenciar, los puntos de referencia son círculos, mientras que los puntos imagen son cuadrillos.

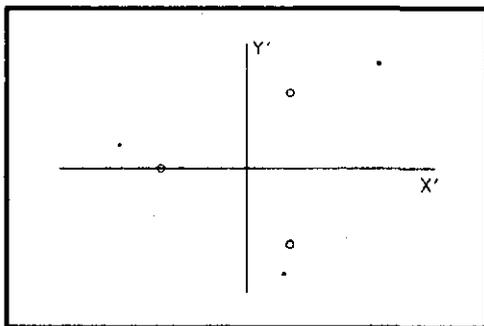


Figura 5.12 Triángulo de Referencia.

La propuesta implica que los puntos imagen tengan que coincidir con los puntos de referencia. Para esto es necesario manipular al hexápodo, de tal manera que el triángulo móvil sea paralelo al plano imagen, con el centro del triángulo coincidente con el eje óptico y además orientado con la disposición que se le dio al triángulo de referencia.

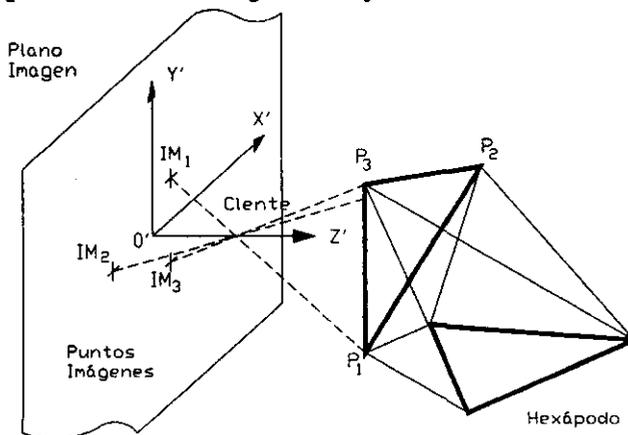


Figura 5.13 Esquema que muestra al hexápodo orientado

La siguiente tabla permite dar una correspondencia e identificación entre los elementos y las variables del sistema.

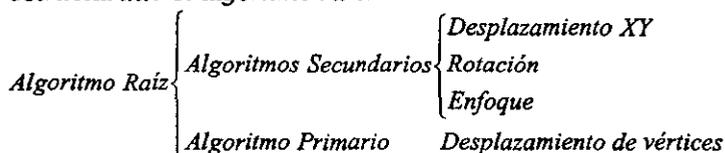
Hexápodo	Color de la imagen	Actuadores
Vértice 1	Rojo	Act. 3 y Act. 4
Vértice 2	Verde	Act. 1 y Act. 2
Vértice 3	Azul	Act. 5 y Act. 6

De acuerdo con esta tabla el vértice 2 se puede identificar en el monitor con el color verde, teniendo como actuadores el 1 y el 2.

Entre los problemas que fue necesario resolver estuvo la generación de movimientos, ya que cada actuador tiene la libertad de realizar sus movimientos en forma independiente de los demás, pero si se cambia la longitud de uno, las coordenadas de los tres vértices del triángulo móvil van a modificarse. Así que si se busca acercar un punto imagen al punto de referencia correspondiente, es muy probable que los otros puntos imágenes se alejen de sus respectivos puntos de referencia. Por está razón fue necesario generar **movimientos tales que permitan reconocer la posición de los vértices en cualquier instante**. Estos movimientos se pueden considerar como **movimientos básicos**, ya que se realizan en función del sistema de referencia del hexápodo (0°). Para esta propuesta se desarrollaron cuatro movimientos:

- Desplazamiento XY
- Rotación
- Enfoque
- Desplazamiento de los vértices

Los primeros tres movimientos son precisamente los algoritmos que se han denominado como secundarios. Son secundarios porque su objetivo se limita en realizar movimientos particulares, cuya finalidad no es la de alcanzar la alineación del espejo, sino que estos movimientos se cumplan para ayudar al algoritmo primario a que no entre en una divergencia o, en su caso, para acelerar el proceso. El último movimiento, "Desplazamiento de los vértices", por ser el más importante se le denomina: algoritmo primario. Su importancia radica en que es el algoritmo fundamental para aproximar la alineación entre los dos espejos. El orden está dado de esa manera, porque así está estructurado el algoritmo raíz.



### 5.6.1 Desplazamiento XY

El objetivo del Desplazamiento XY es el de trasladar el centro del triángulo que se forma con los puntos imágenes (IMc) al centro del triángulo de referencia, o el origen del sistema de referencia  $O'$ . Para lograr este propósito sólo se pueden manipular las longitudes de los actuadores. Esto implica que los vértices del triángulo móvil cambien de posición. El concepto de este movimiento radica en una translación de la plataforma móvil del hexápodo sobre su mismo plano, o sea el plano  $X^*Y^*$ . Esta translación provoca que los puntos imágenes sobre el plano imagen cambien de posición y, para lograr el objetivo de converger los dos triángulos, hay que dirigir esta translación. La dirección se logra determinando el vector ( $V_d$ ) que refiere al centro del triángulo imagen con el origen del sistema  $O'$ .

El vector que define al centro del triángulo imagen, se ve representado en la figura 5.14.

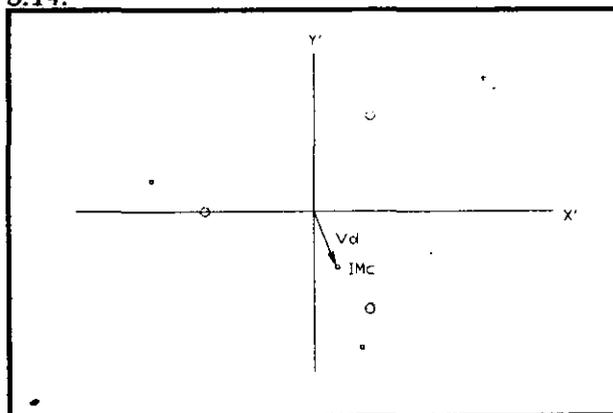


Figura 5.14 Vector que define al centro del triángulo imagen

Debido a que la translación se realiza en el sistema de referencia del hexápodo, mientras que el vector está referido al sistema del plano imagen, y que se desconoce la relación entre los dos sistemas, resulta que la convergencia de los dos triángulos no se puede efectuar de manera directa, sino que es necesario realizarla por medio de aproximaciones.

La condición que permite asegurar la convergencia es cuando la magnitud del vector cumpla con  $\|V_d\| < 1 \times 10^{-7}$

Alcanzada la convergencia en el plano imagen, se obtienen, en correspondencia, las nuevas coordenadas de los vértices del triángulo móvil con respecto al fijo. Se aplica la Cinemática Inversa y se obtiene la longitud que debe alcanzar cada actuador.

### Análisis matemático del Desplazamiento XY.

El vector que define al centro del triángulo imagen es  $Vd = (Vd_x, Vd_y)$ , cuyas componentes y magnitud están dados por:

$$Vd_x = \frac{\sum_{i=1}^3 IMi_x}{3} = \frac{(IM_{1,x} + IM_{2,x} + IM_{3,x})}{3} = IM_{c,x} \quad (5.9 a)$$

$$Vd_y = \frac{\sum_{i=1}^3 IMi_y}{3} = \frac{(IM_{1,y} + IM_{2,y} + IM_{3,y})}{3} = IM_{c,y} \quad (5.9 b)$$

$$|Vd| = \sqrt{(Vd_x)^2 + (Vd_y)^2}$$

Estas componentes se trasladan al sistema de referencia del hexápodo, para generar de esta forma la dirección que permitirá a la plataforma móvil del hexápodo trasladarse sobre su mismo plano. Pero hay que tener cuidado de las unidades, ya que  $Vd_x$  y  $Vd_y$  se encuentran en píxeles. Para asegurar que el ciclo converja más rápido, es conveniente definir un factor  $\beta$  que permita una correspondencia entre el sistema de referencia del plano imagen y el sistema del triángulo móvil.

$$\beta = \frac{LM}{LMi}$$

Al multiplicar  $\beta$  por las componentes del vector  $Vd$ , les asignará las dimensiones correspondientes al sistema  $O^*$ .

Los nuevos valores que adquieren los vértices del triángulo móvil parten de la siguiente expresión:

$$P_i = (P_i)_0 + \Delta P_i \quad (5.10)$$

$(i=1,2,3)$

donde  $P_i$  será el nuevo vector de posición del vértice  $i$ , con respecto al sistema de referencia del triángulo fijo  $O$ .

$(P_i)_0$  es el vector de posición del vértice  $i$ , con respecto a  $O$ .

$\Delta P_i$  es el incremento que sufre el vector de posición, con respecto al sistema  $O$ .

Para determinar el incremento  $\Delta P_i$ , se realiza un producto cruz entre el desplazamiento aproximado que debe realizar el hexápodo ( $Vd$ ) y la matriz de rotación ( $R$ ) que refiere a la plataforma móvil con la fija. De esta manera se alcanza la traslación del hexápodo a lo largo del plano  $X^*Y^*$ .

Esta operación quedaría:

$$\Delta P_i = Vd \times {}^oR = (Vd_x, Vd_y, 0) \times \begin{pmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{pmatrix} \quad (5.11)$$

Sustituyendo la ecuación 5.32 en la 5.31 se obtienen las ecuaciones que definen las nuevas posiciones de los vértices del triángulo móvil con respecto al fijo.

$$P_{ix} = (P_{ix})_0 + Vd_x \cdot xx^* + Vd_y \cdot xy^* \quad (5.12 a)$$

$$P_{iy} = (P_{iy})_0 + Vd_x \cdot yx^* + Vd_y \cdot yy^* \quad (5.12 b)$$

$$P_{iz} = (P_{iz})_0 + Vd_x \cdot zx^* + Vd_y \cdot zy^* \quad (5.12 c)$$

donde  $Vd_x$  y  $Vd_y$  están multiplicados por el factor  $\beta$ .

La longitud de los actuadores se obtiene mediante la Cinemática Inversa:

$$Li = \sqrt{\left({}^{nf}P_{nm}x\right)^2 + \left({}^{nf}P_{nm}y\right)^2 + \left({}^{nf}P_{nm}z\right)^2}$$

Donde  ${}^{nf}P_{nm}$  es el vector que se describe entre los puntos del triángulo fijo y los puntos del triángulo móvil, para "nf" (4,5 y 6) y "nm" (1,2, y 3). Esta fórmula se explica con mayor detalle en el Apéndice A.

### 5.6.2 Rotación

La rotación es el segundo algoritmo de control; este movimiento se realiza alrededor del eje  $Z^*$  del triángulo móvil del hexápodo. Cuando se trató el tema de cómo influyen los seis grados de libertad en el espejo, se hizo notar que la rotación alrededor de  $Z^*$  (giro) no afecta, debido a la simetría del espejo secundario.

A pesar de ser un grado de libertad que no es indispensable controlar, en el caso de esta propuesta de control, que requiere un "Triángulo de Referencia", resulta importante. La razón se debe a que la propuesta busca converger a los puntos imágenes con los puntos de referencia y esto implica que se tenga que girar el hexápodo alrededor del eje  $Z^*$ .

El objetivo, entonces, es girar el hexápodo en torno al eje  $Z^*$ , con el propósito de acercar angularmente los puntos imágenes a los puntos de referencia. Para lograrlo, únicamente se puede manipular la longitud de los actuadores. Haciendo un análisis de la imagen que se presenta en el monitor se deduce que mediante las coordenadas de los puntos imágenes se pueden determinar los ángulos  $ang_i$ , (como se ve en la figura 5.15).

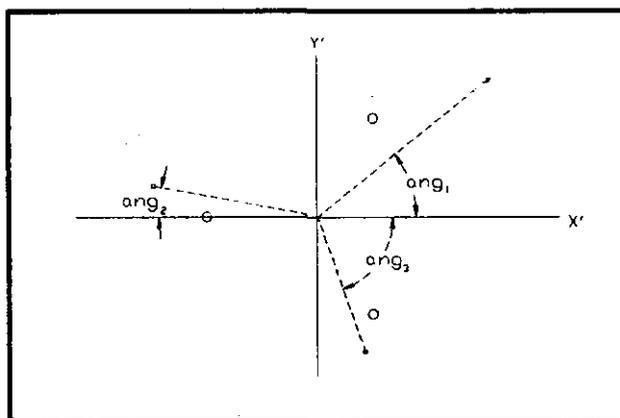


Figura 5.15 Ángulos de las imágenes con respecto al eje  $X'$ .

La razón de haber seleccionado los ángulos con respecto al eje  $X'$  se debe al hecho que estos ángulos se obtienen como resultado del arcotangente. Recordemos la limitante que tiene el manejo de la tangente: en el I<sup>er</sup> y III<sup>er</sup> cuadrante los ángulos son positivos ( $0^\circ$  a  $90^\circ$ ), mientras que en el II<sup>do</sup> y IV<sup>to</sup> cuadrante son negativos ( $-90^\circ$  a  $0^\circ$ ). Esto significa que al aplicar el arcotangente el resultado puede ser válido tanto para un cuadrante como para su complementario.

Esta limitante fue aprovechada para que el resultado final justificara lo siguiente: los ángulos de las imágenes de los vértices (1, 2 y 3) con respecto al eje X, deberán ser  $60^\circ$ ,  $180^\circ$  y  $300^\circ$  respectivamente; pero al introducir el arcotangente los ángulos serán  $60^\circ$ ,  $0^\circ$  y  $-60^\circ$  respectivamente. Este resultado permite realizar una prueba: si se suman los tres ángulos el resultado será nulo. Considerando este ejemplo particular se puede comprender el siguiente concepto: si se suman los tres ángulos y se dividen entre tres, se obtendrá el ángulo promedio, que representa el ángulo que deberá girar el hexápodo para que los puntos imágenes se acerquen lo más posible a los puntos de referencia.

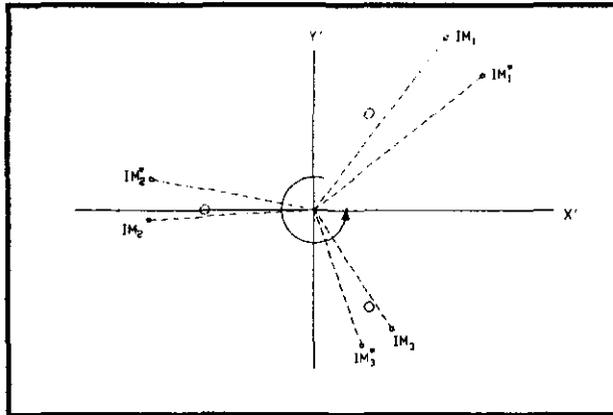


Figura 5.16 Giro de los puntos imágenes, un ángulo promedio.

Al girar los puntos un ángulo promedio, se alcanzarán las nuevas coordenadas de los vértices del hexápodo.

A diferencia del algoritmo del Desplazamiento en XY, aquí no se debe considerar ningún factor, ya que el ángulo es independientemente del tamaño y va a mantenerse constante.

Finalmente, teniendo las coordenadas de los vértices, se puede aplicar la Cinemática Inversa y se obtiene la longitud que debe alcanzar cada actuador

Al igual que el algoritmo anterior, el análisis matemático se realiza de forma inversa a lo que ocurre en la realidad, puesto que el único control físico que se tiene es el cambio longitudinal de los actuadores y lo que se ha realizado hasta el momento es que, a partir de un movimiento básico, se determina la longitud que deben alcanzar los actuadores.

### Análisis matemático de la Rotación en Z.

Para desarrollar este análisis matemático se tienen que determinar los valores de los ángulos ( $ang_i$ )

$$ang_i = \arctan\left(\frac{IMi_y}{IMi_x}\right)$$

Como se explicó anteriormente son valores que oscilan entre  $-90^\circ$  y  $90^\circ$ , realizando el promedio.

$$ang_p = \frac{\sum_{i=1}^3 ang_i}{3} = \frac{ang_1 + ang_2 + ang_3}{3} \quad (5.13)$$

Retomando los conceptos de vector de posición y matriz de rotación, definidos en la Cinemática Directa.

$$p = \left( \frac{(P_{1x} + P_{2x} + P_{3x})}{3}, \frac{(P_{1y} + P_{2y} + P_{3y})}{3}, \frac{(P_{1z} + P_{2z} + P_{3z})}{3} \right) \quad {}_o^o R = \begin{pmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{pmatrix}$$

Las nuevas coordenadas de los vértices ( $P_i$ ) se obtienen a partir de la matriz de rotación, el ángulo de rotación  $ang_p$ , las componentes que definen a los vértices con respecto al sistema de referencia  $O^*$  y del vector de posición.

$$P_1 = -\frac{LM}{2\sqrt{3}} \cdot \begin{bmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{bmatrix} \begin{bmatrix} \cos(ang_p) \\ \sin(ang_p) \\ 0 \end{bmatrix} - \frac{LM}{2} \cdot \begin{bmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{bmatrix} \begin{bmatrix} \sin(ang_p) \\ \cos(ang_p) \\ 0 \end{bmatrix} + p \quad (5.14 a)$$

$$P_2 = \frac{LM}{\sqrt{3}} \cdot \begin{bmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{bmatrix} \begin{bmatrix} \cos(ang_p) \\ \sin(ang_p) \\ 0 \end{bmatrix} + p \quad (5.14 b)$$

$$P_3 = -\frac{LM}{2\sqrt{3}} \cdot \begin{bmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{bmatrix} \begin{bmatrix} \cos(ang_p) \\ \sin(ang_p) \\ 0 \end{bmatrix} + \frac{LM}{2} \cdot \begin{bmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{bmatrix} \begin{bmatrix} \sin(ang_p) \\ \cos(ang_p) \\ 0 \end{bmatrix} + p \quad (5.14 c)$$

Finalmente se obtienen las longitudes de los actuadores (Apéndice A).

### 5.6.3 Enfoque

Se entiende por enfoque, al movimiento del espejo secundario a lo largo del eje  $Z^*$ , con el fin de alcanzar una nitidez en la imagen. Este movimiento se aplica en cualquier equipo óptico, telescopio, binoculares, cámara fotográfica, etc. Para el caso de la simulación no se pretende alcanzar una nitidez, ya que su finalidad es la de alinear a los espejos. No obstante, si se aplica este movimiento en la simulación, la imagen de los leds se mantendrá constante (puntual), aunque el triángulo imagen variará en tamaño dependiendo de la distancia.

Cuando se explicó como se genera la simulación, se mencionó que la distancia deseada entre el espejo secundario y el plano imagen de la cámara iba a definirse como una variable de entrada, y esta distancia ( $2LM$ ) se consideraría como la distancia que debe mantenerse invariante a lo largo de las observaciones telescópicas.

La idea de trasladar al hexápodo a lo largo de del eje  $Z^*$ , permite modificar el tamaño del triángulo que se genera con los puntos imágenes. Con este concepto se puede mencionar que el objetivo de este algoritmo es el de acercar a los puntos imagen con respecto a los puntos de referencia, en función de la diferencia de tamaños que presenta cada triángulo (referencia e imagen). Dicho de otra forma, puesto que el triángulo de referencia es constante, con cada lado de 192 pixeles, mientras que el triángulo imagen tiene los lados variables en función de la posición del triángulo móvil, el propósito de este algoritmo es aproximar el tamaño de ambos triángulos. Es muy probable que la comparación sea entre un triángulo equilátero y uno escaleno, así que la aproximación se referirá a una similitud en tamaño. Para lograr esta aproximación, se efectúa la traslación del hexápodo a lo largo del eje  $Z^*$ , hasta que el perímetro del triángulo imagen alcance el valor del perímetro del triángulo de referencia (576 pixeles).

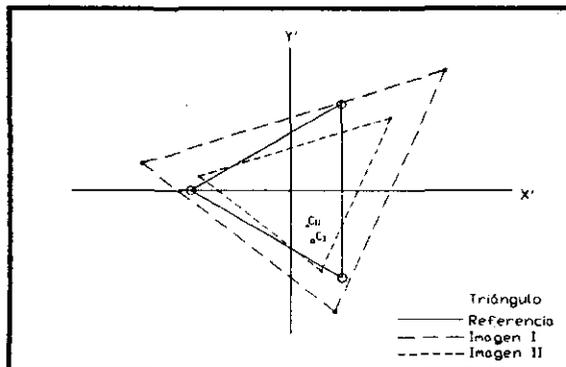


Figura 5.17 Enfoque

### **Análisis matemático del Enfoque.**

El primer paso es determinar el perímetro del triángulo imagen, lo que implica calcular la longitud de cada lado  $L_i$ .

$$L_1 = \sqrt{(IM_2 - IM_3)_x^2 + (IM_2 - IM_3)_y^2}$$

De forma análoga se obtiene  $L_2$  y  $L_3$ . Con la suma de los tres valores se determina el perímetro (Per).

$$Per = L_1 + L_2 + L_3$$

Para determinar el incremento o decremento que debe dársele al enfoque del hexápodo, para aproximar en tamaño al triángulo imagen con respecto al de referencia se hizo el siguiente planteamiento. Si el triángulo imagen fuese perfectamente equilátero tendría un perímetro de 576 píxeles y el hexápodo se encontraría a una distancia de  $2LM$  con respecto al plano imagen; pero si el perímetro es diferente, la distancia entre el hexápodo y el plano imagen será desconocida, aunque se puede considerar la siguiente equivalencia:

$$Dist = \frac{Per * 2LM}{576}$$

De esta forma se obtiene el incremento o decremento, según sea el caso, del enfoque:

$$dEnf = Dist - 2LM \quad (5.15)$$

Las nuevas posiciones de los vértices del triángulo móvil con respecto al fijo son:

$$P_{ix} = (P_{ix})_0 + dEnf \cdot xz^* \quad (5.16 a)$$

$$P_{iy} = (P_{iy})_0 + dEnf \cdot yz^* \quad (5.16 b)$$

$$P_{iz} = (P_{iz})_0 + dEnf \cdot zz^* \quad (5.16 c)$$

$$(i=1,2 \text{ y } 3)$$

Donde  $(P_i)_0$  es el vector de posición del vértice  $i$  con respecto  $O$ .

$P_i$  es el nuevo vector de posición del vértice  $i$ .

$(xz^* \ yz^* \ zz^*)$  son las componentes en  $z^*$  de la matriz de rotación.

Finalmente se obtienen las longitudes de los actuadores de la misma forma que en los algoritmos anteriores. (Ver Apéndice A)

#### 5.6.4 Desplazamiento de los vértices

Este es el algoritmo más importante, en él repercuten directamente el control de todos los grados de libertad. Se puede considerar que los otros algoritmos tienen la función de acelerar el proceso de control, o en su caso romper con un ciclo repetitivo que genera una divergencia.

El planteamiento de este algoritmo es mantener fijos dos vértices del triángulo móvil del hexápodo y mover el tercero. En el monitor de igual manera se verán dos puntos imágenes fijos mientras que otro estará en movimiento. El algoritmo se encuentra en un ciclo y la condición de salida estará dada en el momento en que el punto que se encuentra en movimiento alcanza la distancia mínima al punto de referencia correspondiente. En el momento en que se alcanza este propósito se registra la posición y se cambia el movimiento a otro vértice fijando los dos restantes (incluyendo el que ya sufrió un cambio de posición); el propósito es el mismo, acercar el punto imagen al punto de referencia correspondiente. Finalmente se realiza el mismo procedimiento para el tercer vértice. En el momento en que los tres vértices han realizado sus movimientos en forma independiente se repetirá la operación una y otra vez hasta que el triángulo imagen converja con el triángulo de referencia.

Para poder desarrollar analíticamente el problema de cómo poder mantener fijos dos vértices y manipular sólo uno, se utiliza el siguiente criterio.

Dado un triángulo en el espacio, se fijan dos vértices y se realiza una rotación en torno al vector definido por estos, o sea alrededor del lado del triángulo que contiene dichos vértices.

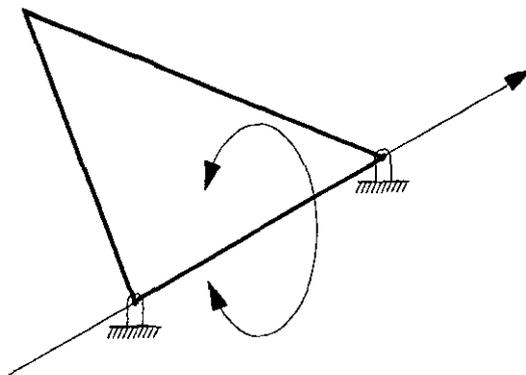


Figura 5.18 Triángulo en el espacio con rotación, manteniendo dos vértices fijos

Se puede apreciar que el vértice móvil realiza un movimiento circular en torno a un punto del vector de rotación. Este punto se puede considerar como el centro de la circunferencia que se forma con el movimiento y se puede determinar trazando un vector del vértice móvil al lado fijo en forma perpendicular. La magnitud del vector es la altura del triángulo

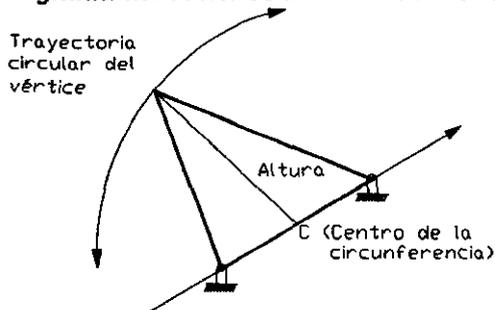


Figura 5.19 Trayectoria del vértice móvil

En función del ángulo que define a la trayectoria circular, se puede trasladar la imagen de un punto en el monitor, manteniendo las otras dos fijas.

#### **Análisis matemático del desplazamiento de los vértices.**

Se puede representar matemáticamente el concepto anterior mediante la geometría analítica.

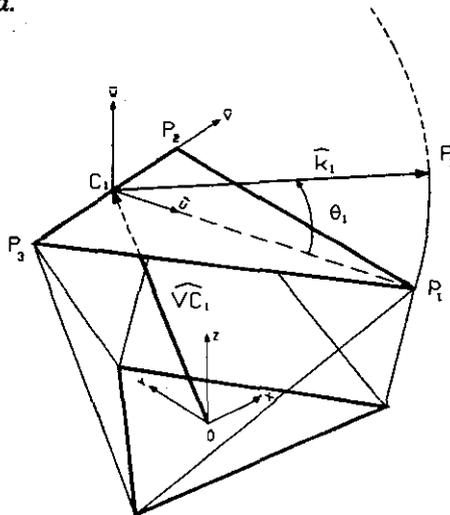


Figura 5.20 Desplazamiento de un vértice en el espacio

ESTA COPIA NO DEBE  
CALR DE LA BIBLIOTECA

El primer cálculo es determinar la posición del centro de la circunferencia que se genera con la trayectoria. La línea que define la altura del triángulo intersecta la base en un punto que representa el centro de la circunferencia, y debido a que el hexápodo está formado por un triángulo equilátero, la altura es equivalente a la mediana. Esto quiere decir que el centro de la circunferencia se encuentra en la mitad de la base, o sea la mitad del lado. Por lo tanto la ecuación que define al centro de la circunferencia del vértice 1 es:

$$C_1 = \frac{P_2 + P_3}{2}$$

Por otro lado se definen los vectores que generan al sistema de referencia  $(u, v, w)$ .

$$\begin{aligned} u_{1x} &= P_{1x} - C_{1x} & v_{1x} &= P_{2x} - C_{1x} \\ u_{1y} &= P_{1y} - C_{1y} & v_{1y} &= P_{2y} - C_{1y} \\ u_{1z} &= P_{1z} - C_{1z} & v_{1z} &= P_{2z} - C_{1z} \end{aligned}$$

Las componentes del vector  $w$  se obtienen del producto cruz de los otros dos vectores.

$$\begin{aligned} w_{1x} &= u_{1y}v_{1z} - u_{1z}v_{1y} \\ w_{1y} &= v_{1x}u_{1z} - v_{1z}u_{1x} \\ w_{1z} &= u_{1x}v_{1y} - u_{1y}v_{1x} \end{aligned}$$

Refiriendo la nueva posición del vértice 1 con respecto al sistema de referencia  $(X, Y, Z)$  del triángulo fijo se tiene.

$${}^0P_{1\theta} = {}^0T_{C_1} C_1P_{1\theta}$$

$${}^0P_{1\theta} = \begin{bmatrix} \frac{u_{1x}}{\|u_1\|} & \frac{v_{1x}}{\|v_1\|} & \frac{w_{1x}}{\|w_1\|} & C_{1x} \\ \frac{u_{1y}}{\|u_1\|} & \frac{v_{1y}}{\|v_1\|} & \frac{w_{1y}}{\|w_1\|} & C_{1y} \\ \frac{u_{1z}}{\|u_1\|} & \frac{v_{1z}}{\|v_1\|} & \frac{w_{1z}}{\|w_1\|} & C_{1z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \|u_1\| \cos \theta_1 \\ 0 \\ \|u_1\| \sin \theta_1 \\ 1 \end{bmatrix} \quad (5.17)$$

La segunda matriz representa al plano donde se está generando la circunferencia; para los tres casos es válido este planteamiento, ya que el sistema de referencia  $(u, v, w)$  se determina para cada caso en particular. Como se ve en la figura 5.20 la circunferencia se genera en el plano  $uw$ , o sea en un plano  $XZ$ , de manera que al multiplicar el módulo de "u", que define al radio de la circunferencia, por las funciones trigonométricas correspondientes, se obtendrá la proyección del vector  $k_1$  en  $u$  y  $w$ .

El ángulo  $\theta_1$  es la variable que va a ir cambiando en este algoritmo, como se verá más adelante. Con esta ecuación queda determinada la nueva posición del vértice  $P_1$ , por lo que finalmente resta calcular la variación de longitud de los actuadores  $L_3$  y  $L_4$  (ver Apéndice A).

El algoritmo se encuentra en un ciclo donde la variable  $\theta_1$  es la que permite modificar la posición del hexápodo. En el monitor de la computadora esta variación se presenta como una curva que parte desde el punto en donde se encuentra la imagen del vértice y se va acercando hacia el vértice de referencia correspondiente. En el momento en que la distancia entre el vértice imagen y el vértice de referencia es mínima se suspende el ciclo y se registra la posición.

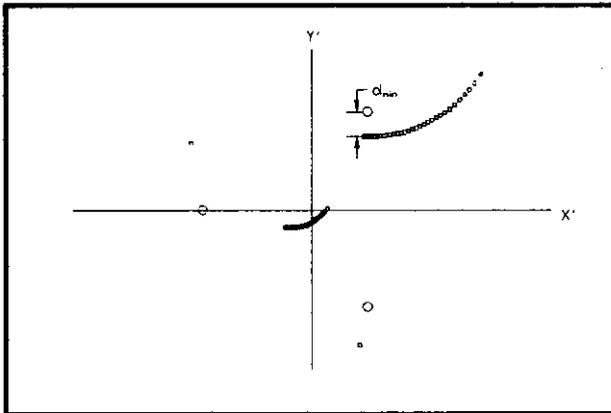


Figura 5.21, Desplazamiento del vértice 1

En la figura 5.21, se puede apreciar como el vértice 1 (rojo), se va acercando al vértice de referencia, con una trayectoria que será función de la inclinación que tenga el hexápodo; por ejemplo, si el hexápodo fuera ortogonal al plano imagen se tendrían trayectorias que se presentarían en forma lineal, y además radial. Por otro lado la trayectoria en negro representa la variación de la posición del centro del triángulo imagen.

Los movimientos de los otros puntos (2 y 3) se obtienen de forma análoga.

La distancia entre el punto imagen y el punto de referencia está dada por:

$$Dist_i = \sqrt{(IM_x - PC_x)^2 + (IM_y - PC_y)^2}$$

La mínima distancia se obtiene mediante una comparación entre el valor de la distancia actual y la anterior.

Desarrollando el producto entre matrices se obtienen los nuevos valores de los vértices del triángulo móvil:

$$P_1 = \frac{a_1}{|a_1|} |\rho_1| \cos \theta_1 + \frac{w_1}{|w_1|} |\rho_1| \sin \theta_1 + C_1 \quad (5.17 a)$$

Finalmente aplicando la Cinemática Inversa se obtienen las nuevas longitudes de los actuadores (ver Apéndice A).

$$L_i = \sqrt{\left({}^{nf} P_{nm} x\right)^2 + \left({}^{nf} P_{nm} y\right)^2 + \left({}^{nf} P_{nm} z\right)^2}$$

Este algoritmo para que pueda converger tiene un "filtro":

- Considerando que se realiza el primer ciclo, la imagen del vértice 1 alcanza la mínima distancia con respecto al punto de referencia correspondiente. Luego se realiza el ciclo para el vértice 2, pero en este caso el movimiento estará dado manteniendo los vértices 1 y 3 fijos, aunque hay que recordar que el vértice 1 ya sufrió un cambio. Se repite el ciclo para el vértice 3, manteniendo como "pivotes" a 1 y 2, los cuales anteriormente han cambiado de posición. Se remarca este hecho porque a la hora de realizar varios ciclos se verá que el vértice 1 converge de forma excelente, pero el vértice 2 ya no tanto, mientras que el vértice 3 estará colocado en una posición en la que nunca convergerá. Esto se debe al hecho que no se mantiene una secuencia igual para todos los vértices, o sea se está dando un privilegio al primer vértice, mientras que los otros estarán en función de la posición que alcanza el primero.

Esta divergencia se rompe con el siguiente "filtro": si al cambiar la posición del vértice 1 solamente se registra su nueva posición, y al realizar el ciclo del vértice 2 y 3 se regresa, el vértice 1, a la posición que tenía inicialmente, se estará dando el mismo peso a cada movimiento. Al concluir con los tres movimientos se sustituyen los tres valores registrados

### 5.6.5 Análisis del Algoritmo Raíz (Triángulo de Referencia)

El Algoritmo Raíz se desarrolla en un ciclo, que se repetirá hasta que la condición de salida sea la alineación de los dos espejos o, en este caso particular, la convergencia del triángulo imagen con el de referencia. Al definir el algoritmo se mencionó que se requiere de un orden de pasos, que el algoritmo raíz presenta de la siguiente manera:

1. Algoritmos secundarios
  - Desplazamiento XY
  - Rotación en Z
  - Enfoque
2. Algoritmo Primario
  - Desplazamiento de los vértices
3. Función matemática
  - Media Pesada

El orden se realiza con esta estructura, pero con una consideración: "Aplicar los algoritmos en cada ciclo puede conllevar a dos cuestiones: primero que si se aplican siempre, o sea en cada ciclo, puede resultar una pérdida de tiempo, pues no siempre son necesarios; y segundo, que en algunos casos puede resultar que dos algoritmos en serie presenten una divergencia o simplemente se contraponen de tal manera que el propósito de alcanzar la convergencia dilate aún más"

Para poder determinar la presencia de cada algoritmo en un ciclo se realizaron diferentes pruebas experimentales. Estas pruebas se explican con mayor detalle en el capítulo de Resultados. Los resultados de las pruebas se muestran en la siguiente tabla:

Algoritmos	Ciclos en que se presentan
Desplazamiento XY	Siempre
Rotación	4,7,10,13,16,19...
Enfoque	3,6,9,12,15,...,24
Desplazamiento de los vértices	Siempre

### **Media Pesada**

Una vez terminado un ciclo, o sea la realización de los algoritmos que están presentes en dicho ciclo, se realiza una media pesada. Una media pesada es una función matemática que permite expresar un resultado como la relación entre el valor inicial y el valor siguiente de una función. Se incluye con el propósito de ser una herramienta que permita converger al sistema, y se expresa de la siguiente manera:

$$P_i = A \cdot P_i^* + B \cdot \hat{P}_i \quad (5.18)$$

donde  $i = 1, 2 \text{ ó } 3$

$P_i^*$  son las coordenadas (x, y, z) del vértice i, después del ciclo.

$P_i^*$  son las coordenadas (x, y, z) del vértice i, antes del ciclo.

$\hat{P}_i$  serán las nuevas coordenadas (x, y, z) del vértice i.

A y B son factores donde  $A + B = 1$

### **Condición de salida**

La condición de salida que permite asegurar la alineación entre los dos espejos se obtiene comparando la distancia entre los puntos imágenes y sus respectivos puntos de referencia. En el momento que las tres diferencias sean nulas se podrá asegurar la alineación.

$$Dist_i = \sqrt{(IM_{ix} - PC_{ix})^2 + (IM_{iy} - PC_{iy})^2}$$

Para determinar una diferencia general se hace una relación entre las tres distancias de cada punto. Esta relación considera como la suma de los cuadrados, ya que si las diferencias entre los puntos son grandes implicaría que la diferencia general sea aún mayor, pero si las distancias se aproximan a cero, la diferencia general será más próxima a cero.

$$Dif = \sum_{i=1}^3 Dist_i^2 = Dist_1^2 + Dist_2^2 + Dist_3^2$$

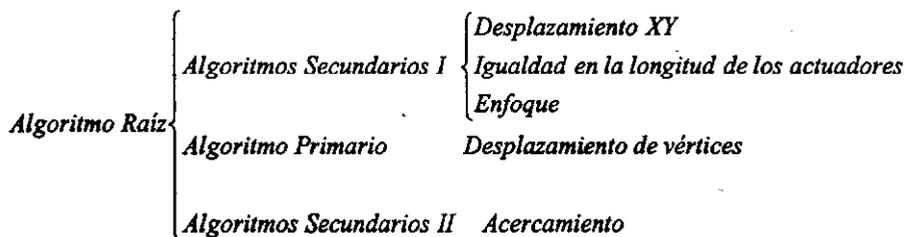
La precisión de la propuesta se acepta para un valor de:

$$Dif < 1 \times 10^{-10}$$



## 5.7 Circunferencia de Referencia

Esta segunda propuesta busca controlar solamente 5 grados de libertad, pues la rotación del espejo alrededor del eje z no interesa. Esta propuesta se genera con una estructura similar a la realizada en la anterior.



El concepto de la propuesta es el siguiente: el triángulo imagen se generará de forma similar a la propuesta anterior con la diferencia que ahora no importa la orientación que presente. Esto significa que la posición se mantendrá en el origen del sistema de referencia (X', Y', Z), así mismo se mantendrá el tamaño del triángulo de 192 pixeles por lado; pero la orientación puede presentarse con cualquier ángulo, de manera que los posibles puntos del triángulo imagen generan una circunferencia.

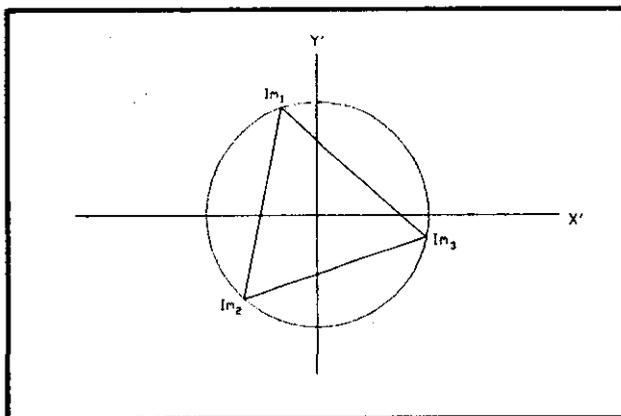


Figura 5.22 Circunferencia de Referencia

El triángulo imagen estará circunscrito en la circunferencia, cuyo radio es la distancia que hay entre cualquier vértice del triángulo imagen y el centro del mismo:

$$rad = \frac{LMi}{\sqrt{3}}$$

### 5.7.1 Algoritmos secundarios I

**Desplazamiento XY.**- El Desplazamiento XY es el mismo que se utilizó en la propuesta del Triángulo de Referencia.

Retomando las ecuaciones expuestas en la sección (5.6.1) las nuevas posiciones de los vértices son:

$$P_{ix} = (P_{ix})_0 + Vd_x \cdot xx' + Vd_y \cdot xy'$$

$$P_{iy} = (P_{iy})_0 + Vd_x \cdot yx' + Vd_y \cdot yy'$$

$$P_{iz} = (P_{iz})_0 + Vd_x \cdot zx' + Vd_y \cdot zy'$$

(i=1,2 y 3)

Se calcula la longitud de cada actuador, aplicando la Cinemática Inversa (ver Apéndice A).

$$L_i = \sqrt{\left({}^{nf}P_{nm}x\right)^2 + \left({}^{nf}P_{nm}y\right)^2 + \left({}^{nf}P_{nm}z\right)^2} ; \text{ donde } i = 1, 2, 3, 4, 5 \text{ y } 6$$

$$nf = 4, 5 \text{ y } 6 \quad nm = 1, 2 \text{ y } 3$$

**Igualdad en la Longitud de los Actuadores.**- A diferencia de la propuesta anterior, en ésta se cambia el algoritmo de Rotación en Z' por el de la Igualdad en la Longitud de los Actuadores. Este algoritmo tiene la función de orientar al triángulo móvil en la posición más próxima del triángulo fijo. Para lograrlo hay que cumplir con la condición que "la suma de las longitudes de los actuadores nones sea igual a la de los pares".

$$\sum L_{non} = \sum L_{par}$$

Se puede considerar como un caso particular de la Rotación en Z'. Por esta razón se aplica el mismo criterio matemático de hacer rotar un ángulo; sólo que en este caso es un ángulo variable (ang), a diferencia de un ángulo promedio de la propuesta anterior (ver 5.6.2 Rotación en Z'). El ángulo se incrementará o decrementará gracias a que el algoritmo se encuentra en un ciclo que permite comparar el estado actual con el anterior, y la condición que permite alcanzar el valor correcto del ángulo, y concluir con el ciclo, es que se satisfaga la ecuación de la igualdad en la longitud de los actuadores.

Por tanto las ecuaciones que definen a los vértices quedan de la siguiente manera.

$$P_1 = -\frac{LM}{2\sqrt{3}} \cdot \begin{bmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{bmatrix} \begin{bmatrix} \cos(ang) \\ \text{sen}(ang) \\ 0 \end{bmatrix} - \frac{LM}{2} \cdot \begin{bmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{bmatrix} \begin{bmatrix} \text{sen}(ang) \\ \cos(ang) \\ 0 \end{bmatrix} + p \quad (5.19 a)$$

$$P_2 = \frac{LM}{\sqrt{3}} \cdot \begin{bmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{bmatrix} \begin{bmatrix} \cos(ang) \\ \text{sen}(ang) \\ 0 \end{bmatrix} + p \quad (5.19 b)$$

$$P_3 = -\frac{LM}{2\sqrt{3}} \cdot \begin{bmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{bmatrix} \begin{bmatrix} \cos(ang) \\ \text{sen}(ang) \\ 0 \end{bmatrix} + \frac{LM}{2} \cdot \begin{bmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{bmatrix} \begin{bmatrix} \text{sen}(ang) \\ \cos(ang) \\ 0 \end{bmatrix} + p \quad (5.19 c)$$

Finalmente las longitudes de los actuadores se obtienen de la misma forma que en el algoritmo anterior (ver Apéndice A).

$$L_i = \sqrt{\left({}^{nf}P_{nm}x\right)^2 + \left({}^{nf}P_{nm}y\right)^2 + \left({}^{nf}P_{nm}z\right)^2} \quad ; \quad \text{donde } i = 1, 2, 3, 4, 5 \text{ y } 6 \\ nf = 4, 5 \text{ y } 6 \quad nm = 1, 2 \text{ y } 3$$

**Enfoque.**- El enfoque, al igual que el Desplazamiento en XY, mantiene la misma estructura matemática de la propuesta anterior.

Como se vio en la sección (5.6.3 Enfoque), las expresiones de los nuevos puntos que definen a los vértices del triángulo móvil del hexápodo quedarían:

$$\begin{aligned} P_{ix} &= (P_{ix})_0 + d\text{Enf} \cdot xz^* \\ P_{iy} &= (P_{iy})_0 + d\text{Enf} \cdot yz^* \\ P_{iz} &= (P_{iz})_0 + d\text{Enf} \cdot zz^* \end{aligned} \quad (i=1, 2 \text{ y } 3)$$

Finalmente ocupando la Cinemática Inversa se obtienen las nuevas longitudes de los actuadores (ver Apéndice A).

$$L_i = \sqrt{\left({}^{nf}P_{nm}x\right)^2 + \left({}^{nf}P_{nm}y\right)^2 + \left({}^{nf}P_{nm}z\right)^2} \quad ; \quad \text{donde } i = 1, 2, 3, 4, 5 \text{ y } 6 \\ nf = 4, 5 \text{ y } 6 \quad nm = 1, 2 \text{ y } 3$$

### 5.7.2 Algoritmo primario

El algoritmo primario o desplazamiento de los vértices, tiene el mismo principio que la propuesta anterior: fijar dos vértices y mover el tercero. Para lograrlo se varía el ángulo  $\theta$  de la misma forma.

Para cambiar un poco el desarrollo matemático, en esta propuesta se cambió la condición de salida. En la propuesta del Triángulo de Referencia la condición que permitía salir del ciclo era la **mínima distancia** entre el punto imagen y su punto de referencia correspondiente. Para esta propuesta se ha desarrollado una nueva alternativa de condición de salida o de satisfacción: la **equidistancia** entre el punto imagen en movimiento y cada uno de los puntos imágenes que se mantienen fijos.

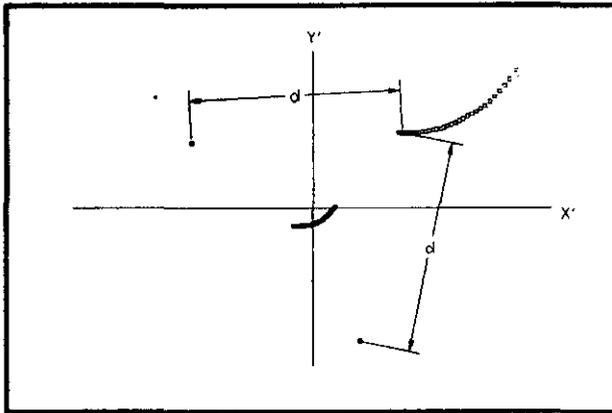


Figura 5.23 Algoritmo primario, equidistancia.

En la figura 5.23 se puede apreciar como se desarrolla este algoritmo. Se hace variar la posición del vértice 1 del triángulo móvil en función del ángulo  $\theta_1$ , provocando que la imagen proyectada en el monitor del mismo vértice cambie su posición. En el momento que se alcanza la equidistancia entre el punto imagen en movimiento ( $IM_1$ ) y los puntos imágenes fijos ( $IM_2$  é  $IM_3$ ), se detiene el ciclo y se repite la misma operación para los otros dos vértices.

La equidistancia está dada por una comparación de los lados del triángulo imagen. Los lados se obtienen de la siguiente manera:

$$Lad_1 = \sqrt{(IM_{2x} - IM_{3x})^2 + (IM_{2y} - IM_{3y})^2}$$

De forma análoga se obtienen los valores de los lados 2 y 3, en donde  $Lad_i$  es el lado opuesto al punto imagen  $IM_i$ .

La posición de los nuevos valores de los vértices del triángulo móvil se obtienen de la misma forma que en la propuesta anterior:

$$P_1 = \frac{a_1}{|w_1|} |p_1| \cos \theta + \frac{w_1}{|w_1|} |p_1| \sin \theta + C_1$$

Aplicando la Cinemática Inversa se obtienen las nuevas longitudes de los actuadores (ver Apéndice A).

$$L_i = \sqrt{\left({}^{nf}P_{nm}x\right)^2 + \left({}^{nf}P_{nm}y\right)^2 + \left({}^{nf}P_{nm}z\right)^2}$$

En esta propuesta aplicamos el mismo "filtro de convergencia" que en la anterior.

### 5.7.3 Algoritmos secundarios II

**Acercamiento.**- Debido a que el algoritmo primario emplea como condición de comparación o salida a la equidistancia, es necesario introducir un nuevo algoritmo, el Acercamiento. La necesidad de aplicar este algoritmo se da porque la condición de equidistancia puede llevar al control a una divergencia, ya que al alcanzar una configuración de actuadores tal que, el triángulo imagen tiende a ser equilátero, la iteración del ángulo  $\theta_i$  provoca que para cualquier nueva posición que tomen los puntos imágenes la condición se cumplirá y como la iteración del ángulo comienza con un paso relativamente grande, entonces es muy probable que después ejecutar el algoritmo, el triángulo imagen reducirá su tamaño.

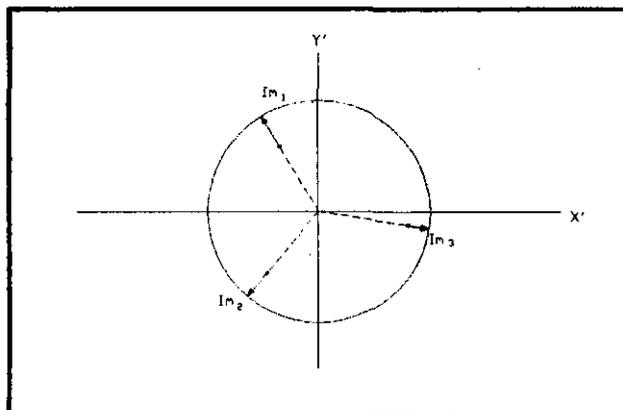


Figura 5.24 Algoritmo de Acercamiento

El algoritmo de acercamiento tiene como principal objetivo trasladar y mantener a los puntos imágenes sobre la circunferencia de referencia. Para lograr esto, se aplica el mismo criterio que se realizó en el algoritmo del Desplazamiento XY, donde se genera un vector, se traslada al sistema O' del hexápodo y se desliza al triángulo móvil con la misma dirección del vector. En este caso se debe realizar el algoritmo tres veces, para cada uno de los vértices. De esta forma es necesario crear un ciclo que permita variar a los vértices. La dirección del vector ( $VD_i$ ) será dada por la normal a la circunferencia que interseca al punto imagen, como es una circunferencia la dirección también es radial, como se ve en la figura 5.24.

El vector ( $VD_i$ ) se obtiene de forma geométrica:

$$VDx_i = \left( \frac{IMi_x}{|VD_i|} \right) \cdot (|VD_i| - rad)$$

$$VDy_i = \left( \frac{IMi_y}{|VD_i|} \right) \cdot (|VD_i| - rad)$$

donde

$$|VD_i| = \sqrt{IMi_x^2 + IMi_y^2}$$

Las ecuaciones que definen a la nueva posición de los vértices son:

$$P_{ix} = (P_{ix})_0 + VDCi_x \cdot xx' + VDCi_y \cdot xy'$$

$$P_{iy} = (P_{iy})_0 + VDCi_x \cdot yx' + VDCi_y \cdot yy'$$

$$P_{iz} = (P_{iz})_0 + VDCi_x \cdot zx' + VDCi_y \cdot zy'$$

( $i=1,2$  y  $3$ )

Se calcula la longitud de cada actuador, aplicando la Cinemática Inversa (ver Apéndice A).

$$Li = \sqrt{\left( {}^{nf}P_{nm}x \right)^2 + \left( {}^{nf}P_{nm}y \right)^2 + \left( {}^{nf}P_{nm}z \right)^2} ; \text{ donde } i = 1, 2, 3, 4, 5 \text{ y } 6$$

$$nf = 4, 5 \text{ y } 6 \quad nm = 1, 2 \text{ y } 3$$

### 5.7.4 Análisis del Algoritmo Raíz (Circunferencia de Referencia)

La estructura del algoritmo raíz quedará de la siguiente manera:

1. Algoritmos secundarios (Desplazamiento XY, Igualdad en la Longitud de los Actuadores y Enfoque)
2. Algoritmo Primario (Desplazamiento de los vértices)
3. Función matemática (Media Pesada)
4. Algoritmo secundario (Acercamiento)

La condición de salida, a diferencia de la propuesta anterior, es la comparación entre la longitud de los lados del triángulo imagen y la longitud final que deben alcanzar, o sea 192 pixeles. Por suma de cuadrados se tiene:

$$Dif = (Lad_1 - 192)^2 + (Lad_2 - 192)^2 + (Lad_3 - 192)^2$$

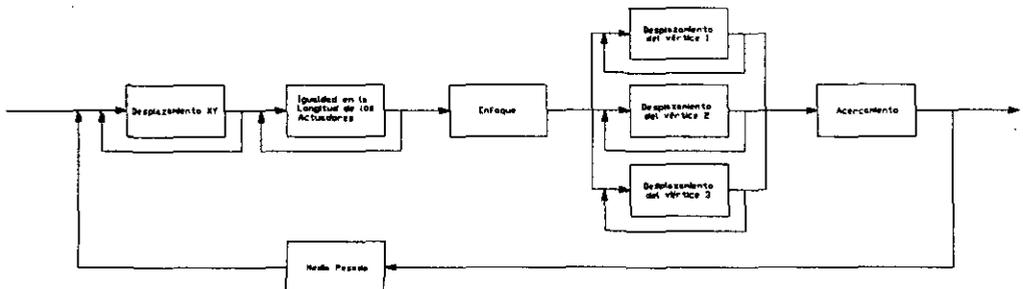
La precisión que alcanza esta propuesta se acepta en:  $Dif < 2 \times 10^{-3}$

Las presencias de los algoritmos son las siguientes:

Algoritmos	Ciclos en que se presentan
Desplazamiento XY	Siempre
Rotación	4,7,10,13,16,19...
Enfoque	Siempre
Desplazamiento de los vértices	Siempre
Acercamiento	Siempre

### 5.7.5 Diagrama de Bloques

Por lo tanto el diagrama de bloques que define a esta propuesta quedaría de la siguiente forma:



## 5.8 Triángulo de referencia con prueba láser

Esta propuesta es idéntica a la del Triángulo de Referencia, solo que incluye una prueba que permite acelerar el proceso. En la aplicación real representa una prueba óptica.

### 5.8.1 Prueba láser

Esta prueba se basa en la inclusión de un haz de luz que va a tener la orientación del eje  $Z^*$ , o sea que parte del centro del espejo secundario y es ortogonal al mismo. En la aplicación real para generar el haz es necesario introducir un láser, el cual se colocará en la estructura que soporta al espejo secundario, de tal manera que para cualquier orientación que presente el espejo el haz será ortogonal a éste.

En el caso de la simulación se considera el láser como una recta que parte del centro del triángulo móvil e intersecta al plano imagen, la dirección será dada por el vector unitario que define al eje  $Z^*$ .

Partiendo de la ecuación general de la recta tenemos

$$IM_{Tmi} = r + tz^* \quad (5.20)$$

donde  $IM_{Tmi}$  es el punto intersección de la recta sobre el plano.

$r$  es el vector de posición del centro del triángulo móvil  $O^*$ .

$z^*$  es el vector unitario del eje  $Z^*$ .

$t$  es el parámetro, un escalar.

El vector unitario se obtiene de la matriz de rotación del triángulo móvil.

$${}^0_o R = \begin{pmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{pmatrix}$$

o sea  $z^* = (xz^*, yz^*, zz^*)$

Mientras que el centro del triángulo móvil esta dado por:

$$r = \left( \frac{(P_{1x} + P_{2x} + P_{3x})}{3}, \frac{(P_{1y} + P_{2y} + P_{3y})}{3}, \frac{(P_{1z} + P_{2z} + P_{3z})}{3} \right)$$

El primer paso es determinar el valor del parámetro de la recta cuando ésta intersecta al plano imagen. En este punto, o punto imagen, la componente en "z" toma un valor nulo y se puede expresar como:

$$IM_{Testz} = 0 = r_z + tz_z^*$$

Despejando

$$t = -\frac{r_z}{z_z^*}$$

Finalmente se obtienen las componentes (x,y) del punto.

$$IM_{Testx} = r_x - \frac{r_z}{z_z^*} z_x^* \quad (5.21 a)$$

$$IM_{Testy} = r_y - \frac{r_z}{z_z^*} z_y^* \quad (5.21 b)$$

### 5.8.2 Análisis de la propuesta

**Algoritmos secundarios.**- Los algoritmos secundarios son idénticos a los utilizados en la propuesta del Triángulo de Referencia.

**Algoritmo primario.**- El algoritmo primario tendrá la misma estructura matemática que la utilizada en la propuesta del Triángulo de Referencia, sólo que se diferenciará por la condición de comparación o salida.

La condición de comparación o salida del algoritmo primario representa la distancia que habrá entre el punto prueba ( $IM_{Test}$ ) y el centro del sistema de referencia.

$$Dist_t = \sqrt{(IM_{Testx} - PC_{Cx})^2 + (IM_{Testy} - PC_{Cy})^2}$$

En el momento que la distancia sea nula se finalizará el algoritmo.

**Algoritmo Raíz.-** El algoritmo Raíz también tendrá la misma estructura utilizada en la primera propuesta. La presencia de los algoritmos será:

Algoritmos	Ciclos en que se presentan
Desplazamiento XY	Siempre
Rotación	4, 7, 10, 13, 16, 19...
Enfoque	3, 6, 9, 12, 15, ..., 24
Desplazamiento de los vértices	Siempre

La condición de salida del algoritmo Raíz es la misma que se maneja en la propuesta del triángulo de Referencia:

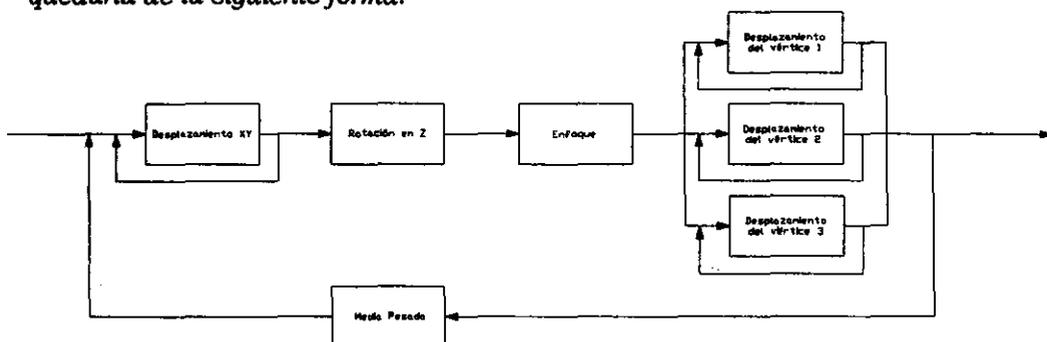
$$Dist_i = \sqrt{(IM_{ix} - PC_{ix})^2 + (IM_{iy} - PC_{iy})^2}$$

$$Dif = \sum_{i=1}^3 Dist_i^2 = Dist_1^2 + Dist_2^2 + Dist_3^2$$

Aunque la precisión de esta propuesta se acepta para  $Dif < 2.2 \times 10^{-3}$

### 5.8.3 Diagrama de Bloques

Por lo tanto el diagrama de bloques que define a esta propuesta quedaría de la siguiente forma:



## 6. RESULTADOS Y COMPROBACIONES

*El objetivo de este capítulo es el de comprobar la validez de las teorías que se desarrollaron a lo largo de este proyecto. Se realizó para cada tema un programa de computadora que permite simular el estado geométrico del hexápodo y los resultados son el desarrollo de la teoría. Las comprobaciones de los programas se pueden dar mediante una configuración especial de los actuadores, donde el resultado final se puede calcular analíticamente, o por medio de programas hechos especialmente para justificar la teoría.*

*Los temas que se trataron en este proyecto son:*

- Cinemática
  - { Directa
  - { Inversa
  - { Generación de trayectorias
- Dinámica
- Control
  - { Triángulo de referencia
  - { Circunferencia de referencia
  - { Triángulo de referencia con prueba láser

*A continuación se describirán las comprobaciones de cada tema.*

## 6.1 Cinemática Directa.

En la Cinemática Directa se obtuvieron los ángulos de Euler comparando algunas celdas entre la matriz de rotación generada y la matriz de rotación de Euler. Al sustituir estos ángulos en las celdas restantes se obtiene numéricamente la matriz de rotación de Euler, esto permite compararla con la matriz de rotación generada y si el valor de cada una de las celdas es igual se puede asegurar el análisis teórico. Este mismo criterio se emplea para la matriz de rotación con vector auxiliar.

A continuación se presenta un ejemplo para demostrar la validez del análisis teórico:

Se consideren las siguientes longitudes de los actuadores y del lado de cada triángulo (fijo y móvil):

$$\begin{array}{lll} L_1 = 22 \text{ u.l.} & L_4 = 19.5 \text{ u.l.} & LM = 25 \text{ u.l.} \\ L_2 = 18 \text{ u.l.} & L_5 = 24 \text{ u.l.} & LF = 21 \text{ u.l.} \\ L_3 = 23 \text{ u.l.} & L_6 = 17 \text{ u.l.} & \end{array}$$

La matriz de rotación generada por el análisis geométrico resulta:

$${}^0R = \begin{pmatrix} xx^* & xy^* & xz^* \\ yx^* & yy^* & yz^* \\ zx^* & zy^* & zz^* \end{pmatrix} = \begin{bmatrix} \frac{P_{2x}'}{|P_2'|} & \frac{C_{23x}}{|C_{23}|} & \frac{K_x}{|K|} \\ \frac{P_{2y}'}{|P_2'|} & \frac{C_{23y}}{|C_{23}|} & \frac{K_y}{|K|} \\ \frac{P_{2z}'}{|P_2'|} & \frac{C_{23z}}{|C_{23}|} & \frac{K_z}{|K|} \end{bmatrix} = \begin{bmatrix} 0.9375888 & 0.3227626 & 0.1294284 \\ -0.3323926 & 0.9411786 & 0.0608104 \\ -0.1021879 & -0.1000362 & 0.9897224 \end{bmatrix}$$

De esta matriz se obtienen los ángulos de Euler:

$$\psi = -5.771571^\circ \quad \theta = 5.865176^\circ \quad \phi = -19.52036^\circ$$

Sustituyendo los valores de los ángulos en la matriz de rotación de Euler, se tiene:

$$R_{\phi,\theta,\psi} = \begin{bmatrix} c\phi c\theta & c\phi s\theta s\psi - s\phi c\psi & c\phi s\theta c\psi + s\phi s\psi \\ s\phi c\theta & s\phi s\theta s\psi + c\phi c\psi & s\phi s\theta c\psi - c\phi s\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{bmatrix} = \begin{bmatrix} 0.9375888 & 0.3227623 & 0.1294284 \\ -0.3323926 & 0.9411787 & 0.0608104 \\ -0.1021879 & -0.1000362 & 0.9897224 \end{bmatrix}$$

De igual forma para obtener las componentes del vector auxiliar y el ángulo de rotación, se compararon algunas celdas de ambas matrices. Los valores resultantes son:

$$\alpha = 20.89351^\circ \quad \hat{k} = (-0.2255077, 0.3247272, -0.9185306)$$

Sustituyendo estos valores se obtiene la matriz numérica.

$$R_{k,\alpha} = \begin{bmatrix} k_x^2 v\alpha + c\alpha & k_x k_y v\alpha - k_x s\alpha & k_x k_z v\alpha + k_y s\alpha \\ k_x k_y v\alpha + k_x s\alpha & k_y^2 v\alpha + c\alpha & k_y k_z v\alpha - k_x s\alpha \\ k_x k_z v\alpha - k_y s\alpha & k_y k_z v\alpha + k_x s\alpha & k_z^2 v\alpha + c\alpha \end{bmatrix} = \begin{bmatrix} 0.9375888 & 0.3227625 & 0.1294284 \\ -0.3323928 & 0.9411786 & 0.0608104 \\ -0.1021879 & -0.1000362 & 0.9897224 \end{bmatrix}$$

Como se puede visualizar las tres matrices numéricas son prácticamente iguales, por lo tanto la teoría realizada para la Cinemática Directa es la correcta.

Las componentes del vector de posición se pueden comprobar mediante ciertas configuraciones especiales de los actuadores, o por medio de la correspondencia que existe entre la Cinemática Directa e Inversa.

El programa proporciona los siguientes valores de las componentes del vector.

$$p = \left( \frac{(P_{1x} + P_{2x} + P_{3x})}{3}, \frac{(P_{1y} + P_{2y} + P_{3y})}{3}, \frac{(P_{1z} + P_{2z} + P_{3z})}{3} \right) = (1.874537, 0.988149, 15.32787)$$

## 6.2 Cinemática Inversa

La Cinemática Inversa se determina mediante una suma vectorial y para justificar su desarrollo teórico es necesario aplicar la correspondencia que existe con la Cinemática Directa. Se desarrollaron dos análisis de Cinemática Inversa: la matriz de rotación de Euler, cuyos parámetros geométricos son  $r_x, r_y, r_z, \psi, \theta, \phi$  y la matriz de rotación con un vector auxiliar, con  $r_x, r_y, r_z, \alpha, k$  como los parámetros geométricos. El programa de computadora tiene como datos de entrada a dichos parámetros geométricos, y el objetivo es determinar la longitud de cada uno de los actuadores.

Aplicando el mismo ejemplo que en la Cinemática Directa, se tiene que para el análisis con la matriz de Euler los valores de entrada son:

$\psi = -5.771571^\circ$	$r = (1.874537, 0.988149, 15.32787)$
$\theta = 5.865176^\circ$	$LM = 25$
$\phi = -19.52036^\circ$	$LF = 21$

Se obtienen los valores de las longitudes de los actuadores:

$L_1 = 22$ u.l.	$L_4 = 19.5$ u.l.
$L_2 = 18$ u.l.	$L_5 = 24$ u.l.
$L_3 = 23$ u.l.	$L_6 = 17$ u.l.

Con esto se comprueba la correspondencia que existe entre la Cinemática Directa y la Cinemática Inversa y, por lo tanto, que el valor de las componentes del vector  $r$  son las correctas.

Para el caso de la matriz con el vector auxiliar, se realiza el mismo ejemplo, donde los datos de entrada son:

$\alpha = 20.89351^\circ$	$r = (1.874537, 0.988149, 15.32787)$
$k = (-0.2255077, 0.3247272, -0.9185306)$	$LM = 25$
	$LF = 21$

Las longitudes que se obtienen del programa son:

$L_1 = 22$ u.l.	$L_4 = 19.5$ u.l.
$L_2 = 18$ u.l.	$L_5 = 24$ u.l.
$L_3 = 23$ u.l.	$L_6 = 17$ u.l.

De igual forma se comprueba que el análisis de la Cinemática Directa y la Inversa son correctos.

### 6.3 Generación de trayectorias

La ecuación que se consideró como una posible función para generar las trayectorias que debe seguir el hexápodo es la siguiente:

$$Li = Li_0 + \frac{Li_f - Li_0}{t_f - t_0}(t - t_0)$$

donde  $Li$  es la longitud del actuador  $i$  en un tiempo  $t$   
 $Li_0$  es la longitud inicial de cada actuador  
 $Li_f$  es la longitud final de cada actuador  
 $t_f$  es el tiempo total para realizar la trayectoria  
 $t_0$  es el tiempo inicial que se puede considerar cero

Para comprobar que la ecuación anterior permite generar las trayectorias, se consideran las condiciones límite:

$$t = t_0$$

$$Li = Li_0 + \frac{Li_f - Li_0}{t_f - t_0}(t_0 - t_0) = Li_0$$

$$t = t_f$$

$$Li = Li_0 + \frac{Li_f - Li_0}{t_f - t_0}(t_f - t_0) = Li_0 + Li_f - Li_0 = Li_f$$

Ambas condiciones proporcionan lo esperado, lo cual indica que la ecuación cumple con su objetivo.

## 6.4 Dinámica

Se considere a un hexápodo con la siguiente configuración:

El lado del triángulo fijo	LF	20 u.l.
El lado del triángulo móvil	LM	10 u.l.
La longitud de cada actuador	$L_i$	15 u.l.

Esta configuración permite hacer el siguiente análisis geométrico: debido a que el lado del triángulo móvil es la mitad del lado del triángulo fijo y que los actuadores son iguales (Figura 6.4.1a), los ángulos  $\theta_i$  ( $i=1,2,3$ ) tendrán un valor de  $90^\circ$ . En la Figura 6.4.1b se muestra la proyección de planta del hexápodo, esquematizando a los actuadores colineales a los lados del triángulo fijo.

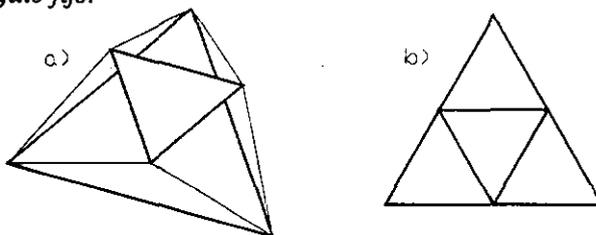


Figura 6.4.1 Configuración especial del hexápodo

Para comprobar la veracidad del método se realizaron tres pruebas:

- Análisis mediante desplazamientos virtuales
- Análisis estático
- Análisis con el desplazamiento de todos los actuadores.

### 6.4.1 Desplazamientos virtuales

El desarrollo de esta prueba es el descrito en el capítulo de la Dinámica, como el desplazamiento de un actuador y se obtiene mediante un programa de computadora. Si se considera que el incremento que sufre el actuador es de 0.01 u.l., se desarrolla primero el programa de la Cinemática Directa para obtener los nuevos vectores de posición de los vértices, para posteriormente aplicar el programa de la Dinámica, el cual muestra el siguiente resultado:

$$f_a = f_g \frac{\Delta_g}{\Delta_a} = f_g \frac{0.0022347}{0.01}$$

$$f_a = 0.22347 f_g \quad (6.4.1)$$

### 6.4.2 Análisis estático

El triángulo generado por los actuadores y un lado del triángulo fijo se presenta en la figura 6.4.2, con las dimensiones establecidas.

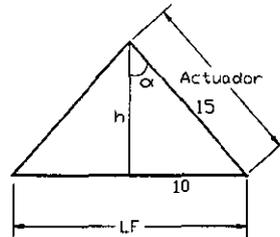


Figura 6.4.2 Triángulo generado por los actuadores, dimensionado.

Por otro lado se puede generar el siguiente diagrama de fuerzas, figura 6.4.3.

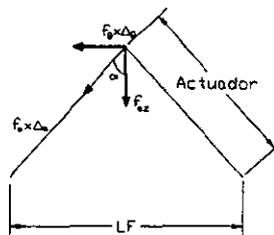


Figura 6.4.3 Diagrama de fuerzas.

De las figuras 6.4.2 y 6.4.3 se puede obtener que:

$$f_a = f_g \cos \alpha = f_g \frac{\sqrt{(15)^2 - (10)^2}}{15} \quad (6.4.2)$$

Debido a que los actuadores tienen la misma longitud, se puede considerar que la fuerza que ejerce cada uno de los actuadores en su componente en z es un sexto de la fuerza de gravedad

$$f_a = \frac{f_z}{6} \quad (6.4.3)$$

Sustituyendo la ecuación 6.4.3 en la ecuación 6.4.2 se obtiene la fuerza que ejerce el actuador:

$$f_a = \frac{f_z}{6 \cdot \frac{\sqrt{(15)^2 - (10)^2}}{15}}$$

$$f_a = 0.223607 f_z \quad (6.4.4)$$

### 6.4.3 Desplazamiento de todos los actuadores.

Del triángulo generado por los actuadores y un lado del triángulo fijo (figura 6.4.2), se puede desarrollar la siguiente ecuación:

$$h = \sqrt{15^2 - 10^2} = 11.18034 \quad (6.4.5)$$

Si se considera que todos los actuadores sufren un incremento de 0.01 u.l, la altura del triángulo cambia su magnitud:

$$h' = \sqrt{15.01^2 - 10^2} = 11.193753 \quad (6.4.6)$$

Realizando la diferencia con el valor obtenido para  $h$  de la ecuación 6.4.5, se tiene que:

$$\Delta h = 11.193753 - 11.18034 = 0.013413$$

Al incrementar todos los actuadores, es equivalente el incremento de  $h$  al del centro del triángulo, además se da en la dirección del eje  $z$ , o sea la dirección de la gravedad, por lo tanto:

$$\Delta_z = \Delta h = 0.013413 \quad (6.4.7)$$

Debido a que se incrementaron los seis actuadores, la ecuación 4.1 se expresaría de la siguiente manera:

$$6 \cdot f_a = f_z \frac{\Delta_z}{\Delta_a} \quad (6.4.8)$$

Sustituyendo la ecuación 6.4.7 en la 6.4.8 se obtiene que:

$$f_a = f_z \frac{.013413}{6 \times 0.01} = 0.22355 f_z$$

$$f_a = 0.22355 f_z \quad (6.4.9)$$

Finalmente, al comparar el resultado obtenido por medio del método de los desplazamientos virtuales en la sección 6.4.1 con los obtenidos en las secciones posteriores, se puede asegurar que el método es confiable.

## 6.5 Control.

Para comprobar la validez del Control se desarrollaron una serie de mediciones de las imágenes como: la longitud de cada lado del triángulo imagen, la posición de cada punto imagen con respecto al sistema de referencia del plano ( $O'$ ) y la diferencia entre los puntos imágenes y los puntos de referencia. Debido a que ninguna medición proporciona información acerca de la perspectiva que tiene la plataforma móvil se incluyó la prueba láser en las tres propuestas, misma prueba que en la tercera propuesta sirve como condición de comparación.

### Presencia de los Algoritmos

La presencia de los algoritmos en los ciclos se determinó mediante una serie de pruebas. Cada prueba se generó con diferentes configuraciones de longitudes de los actuadores y con una matriz de referencia diferente, matriz que referencia al sistema  $O'$  del sistema  $O$ . En cada prueba se varió la presencia de los algoritmos.

Se realizaron aproximadamente unas 30 pruebas diferentes y, para cada propuesta, se expone la presencia de los algoritmos que permite converger a los programas de control con mayor facilidad:

<i>Triángulo de Referencia con y sin láser</i>	
<i>Algoritmos</i>	<i>Ciclos en que se presentan</i>
<i>Desplazamiento XY</i>	<i>Siempre</i>
<i>Rotación</i>	<i>4,7,10,13,16,19...</i>
<i>Enfoque</i>	<i>3,6,9,12,15,...,24</i>
<i>Desplazamiento de los vértices</i>	<i>Siempre</i>

<i>Circunferencia de Referencia</i>	
<i>Algoritmos</i>	<i>Ciclos en que se presentan</i>
<i>Desplazamiento XY</i>	<i>Siempre</i>
<i>Rotación</i>	<i>4,7,10,13,16,19...</i>
<i>Enfoque</i>	<i>Siempre</i>
<i>Desplazamiento de los vértices</i>	<i>Siempre</i>
<i>Acercamiento</i>	<i>Siempre</i>

### Resultados de los programas de las propuestas de control

En los programas de las propuestas de control aparece la simulación de las proyecciones que permite visualizar los nuevos estados del sistema con la inclusión de cada algoritmo. Las mediciones de las imágenes se realizan al efectuar un ciclo completo, esto permite generar gráficas que indican el estado del sistema después de un ciclo.

Debido a que cada propuesta de control alcanza resultados diferentes, por la precisión que proporciona cada una y por el hecho que en cada una de las propuestas la presencia que tienen los algoritmos en cada ciclo es diferente, se va a considerar un mismo ejemplo para las tres propuestas y sobre los resultados se podrán realizar algunas conclusiones.

Sea la configuración del hexápodo la siguiente:

$$\begin{array}{lll}
 L_1 = 25 \text{ u.l.} & L_4 = 23 \text{ u.l.} & LM = 25.4 \text{ u.l.} \\
 L_2 = 15 \text{ u.l.} & L_5 = 19 \text{ u.l.} & LF = 21 \text{ u.l.} \\
 L_3 = 17 \text{ u.l.} & L_6 = 21 \text{ u.l.} &
 \end{array}$$

donde las componentes de la matriz transformación que referencia al sistema  $O'$  del sistema  $O$ , tienen los siguientes valores:

Ángulos de Euler	Vector de posición
$\psi' = -12^\circ$	$r'_x = 32$
$\theta' = 7^\circ$	$r'_y = -15$
$\phi' = -19^\circ$	$r'_z = 63$

Por lo que la matriz quedaría:

$${}^O_T = \begin{bmatrix} c\phi'c\theta' & c\phi's\theta's\psi' - s\phi'c\psi' & c\phi's\theta'c\psi' + s\phi's\psi' & r'_x \\ s\phi'c\theta' & s\phi's\theta's\psi' + c\phi'c\psi' & s\phi's\theta'c\psi' - c\phi's\psi' & r'_y \\ -s\theta' & c\theta's\psi' & c\theta'c\psi' & r'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.9384708 & 0.2944961 & 0.1804011 & 32 \\ -0.3231414 & 0.933106 & 0.1577746 & -15 \\ -0.1218693 & -0.2063619 & 0.9708566 & 63 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De esta forma ya están generados los parámetros que se requieren para poder desarrollar las propuestas de control.

### Triángulo de Referencia.

Desarrollando el control se alcanza la solución del problema después de 40 ciclos; para visualizar lo sucedido en cada uno ellos a continuación se presentan los siguientes gráficos

El control logra la solución cuando los lados del triángulo imagen alcanzan un valor de 192 píxeles (2 pulg.). De esta forma, en el Gráfico 6.1 de "Longitud de los Lados del Triángulo Imagen vs. Ciclos", se puede apreciar como la dimensión de los lados se aproxima en cada ciclo al valor final, hasta que finalmente convergen.

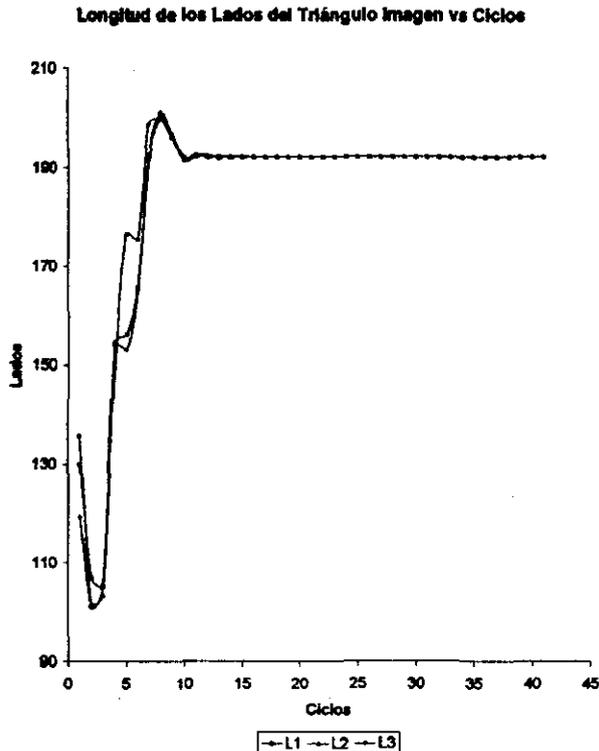


Gráfico 6.1 Longitud de los Lados del triángulo Imagen vs. Ciclos

En el gráfico se puede apreciar como los lados del triángulo imagen tienden a converger en 192 píxeles. Esto significa que el triángulo imagen es perfectamente equilátero. La característica de la curva está dada por la presencia de los algoritmos en cada ciclo. La base de datos con que fue elaborado el gráfico se muestra en el Apéndice B (I-1).

Otra forma de comprobar la convergencia del control es mediante la medición de las distancias entre los vértices del triángulo imagen y los vértices del el triángulo de referencia. En el Gráfico 6.2, "Distancias entre los puntos imágenes y los puntos de referencia correspondientes vs Ciclos", se incluye la distancia entre la imagen de la prueba láser y el origen del sistema O', la cual permite constatar la validez del control.

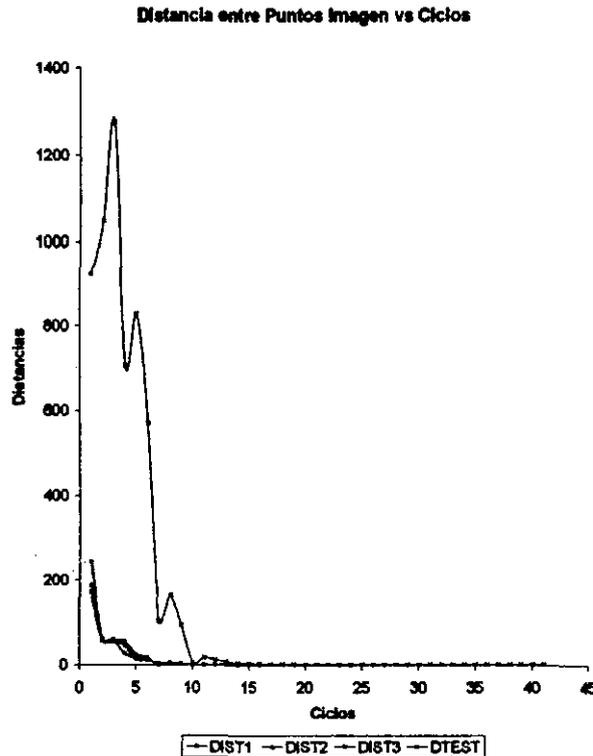


Gráfico 6.2 Distancias entre los puntos imagen y los puntos de referencia vs. Ciclos

Este gráfico permite comprobar como se encuentra la posición del triángulo imagen con respecto al triángulo de referencia. La curva de la prueba del láser es representativa de la convergencia de la propuesta de control, ya que alcanza un valor de 0.612 píxeles con respecto al origen O'. Esta cantidad indica que el láser tiene una desviación angular de:

$$\text{ang} = \tan^{-1}\left(\frac{D_{\text{test}}}{2 \cdot LM}\right) = \tan^{-1}\left(\frac{0.612}{96 \cdot (2 \cdot 10)}\right) = 0.01863^\circ$$

Lo cual representa una desviación de aproximadamente un minuto. La base de datos del gráfico se muestra en el Apéndice B (I-2).

Finalmente se presenta el Gráfico 6.3 que muestra como los actuadores alcanzan su longitud final a lo largo de los ciclos, "Longitud de los actuadores vs. Ciclos".

Este gráfico permite comprobar la veracidad del control, ya que al compararlo con los resultados obtenidos en las otras propuestas se podrá confirmar la veracidad de los resultados.

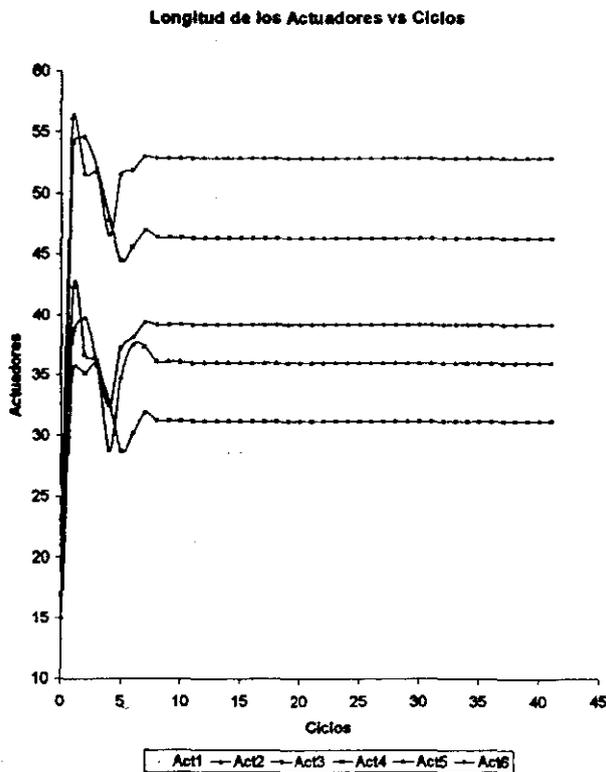


Gráfico 6.3 Longitudes de los Actuadores vs. Ciclos

Los resultados obtenidos en este gráfico muestran como los actuadores alcanzan en pocos ciclos un valor muy aproximado al valor final, esto es indicativo de la gran eficiencia que tiene el control para alcanzar el valor final.

La base de datos que se empleó para generar el gráfico se muestra en el Apéndice B (I-3).

### Circunferencia de Referencia.

Utilizando el mismo ejemplo que se empleó en el caso de la propuesta del Triángulo de Referencia, se alcanza la convergencia después de aproximadamente 18 ciclos; los resultados que se obtuvieron se pueden observar en el siguiente Gráfico 6.4:

Longitud de los Lados del Triángulo Imagen vs Ciclos

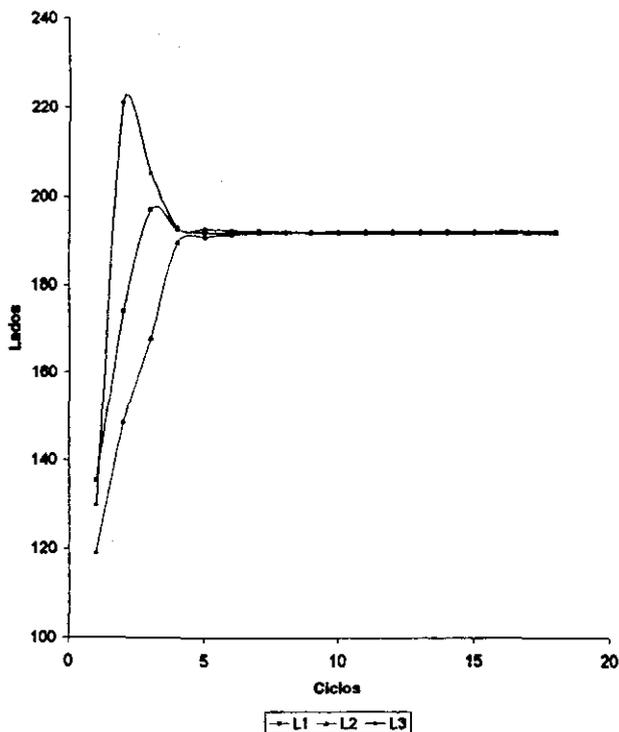


Gráfico 6.4 Longitud de los Lados del triángulo Imagen vs. Ciclos

A pesar de que la longitud de los lados del triángulo imagen no alcanza a converger en 192 píxeles, se puede considerar que los valores tienden a aproximarse a ese valor y por lo tanto el triángulo imagen se puede tomar como equilátero.

Si en esta propuesta se dejara de aplicar el algoritmo de Enfoque (ciclo 25), el sistema comenzaría a divergir; si se alternara cada 3 ciclos, como en las otras propuestas, el sistema comenzaría a oscilar. La importancia de este algoritmo se debe a las limitaciones que presenta el algoritmo primario cuya condición de salida es la equidistancia. La base de datos para generar el gráfico está en el Apéndice B (II - 1).

El Gráfico 6.5, que incluye a las distancias entre los puntos imagen y la circunferencia de referencia, además de la distancia entre la proyección del láser y el centro del sistema de referencia, quedaría de la siguiente manera:

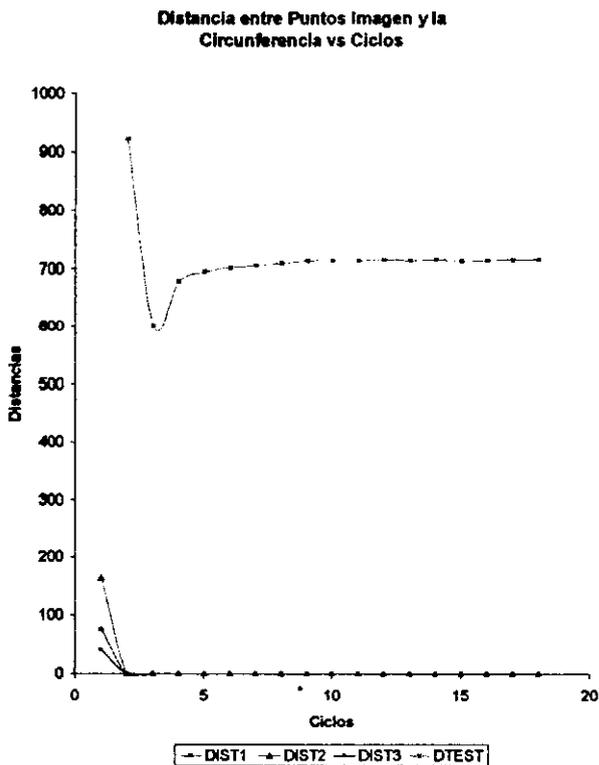


Gráfico 6.5 Distancias entre los puntos imagen y la circunferencia de referencia vs. Ciclos

Debido a que en esta propuesta se incluye el algoritmo del Acercamiento, la distancia entre los puntos imagen y la circunferencia es nula.

Lo interesante de esta gráfica es la curva que se genera con los puntos imagen de la prueba del láser, ya que es indicativo que aunque el resultado converge, la precisión será muy baja con respecto a la obtenida en la propuesta anterior. La base de datos que se empleó para generar esta gráfica se muestra en el Apéndice B (II - 2).

Las longitudes de los actuadores variarán en función de los ciclos, como se ve en el Gráfico 6.6:

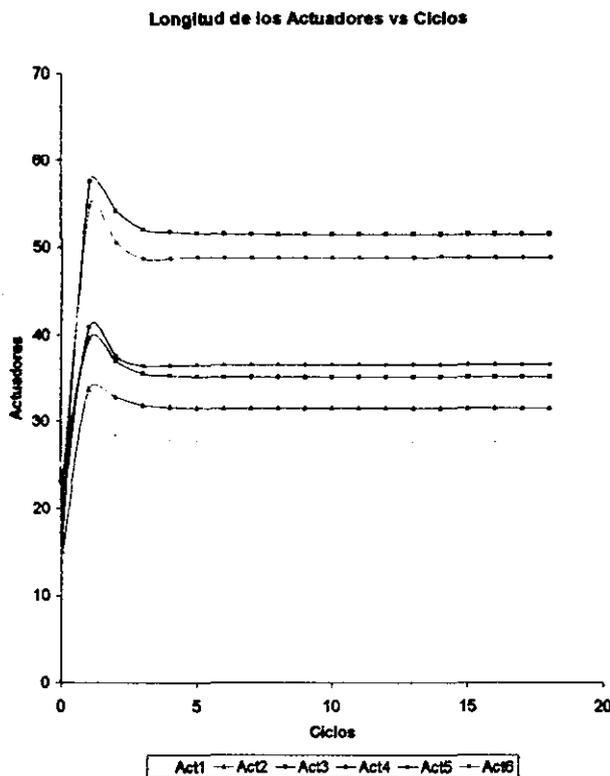


Gráfico 6.6 Longitudes de los Actuadores vs. Ciclos

Con este gráfico no se pueden comparar los resultados que se obtuvieron en las otras propuestas, ya que la precisión que presenta la prueba láser, en el gráfico 6.5, es tan mala que las longitudes de los actuadores no alcanzan el valor deseado para la alineación de los dos espejos.

La base de datos que se empleó para realizar esta gráfica se muestra en el Apéndice B (II - 3).

### Triángulo de Referencia con Láser

La propuesta del Triángulo de Referencia con Láser converge en 18 ciclos. Los resultados se muestran en los siguientes gráficos.

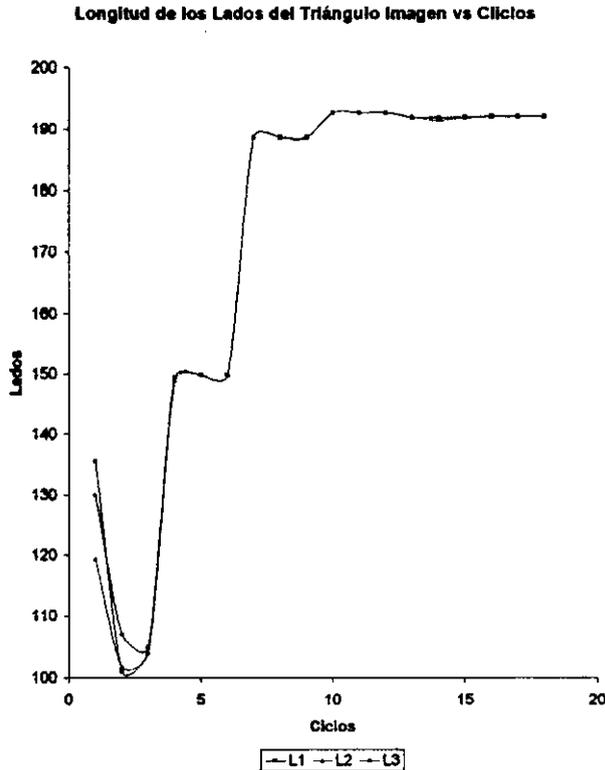


Gráfico 6.7 Longitud de los Lados del triángulo Imagen vs. Ciclos

Los lados del triángulo imagen convergen aproximadamente en 192 píxeles. A pesar de que al aplicar la prueba del láser se espera que los resultados alcancen valores más precisos, en realidad no es así, ya que al introducir la prueba láser como condición de comparación se puede considerar que la propuesta de control se encuentra en un sistema de lazo cerrado. Esto implica que al haber retroalimentación se puede caer en un exceso de condiciones que llevan al sistema a oscilar, sin poder así alcanzar los valores deseados.

La base de datos con que fue elaborado el Gráfico 6.7 se muestra en el Apéndice B (III-1).

El Gráfico 6.8 que corresponde a la distancia entre los puntos imagen y los puntos de referencia, además de la distancia entre la imagen del láser y el centro del sistema de referencia  $O'$ , es el siguiente:

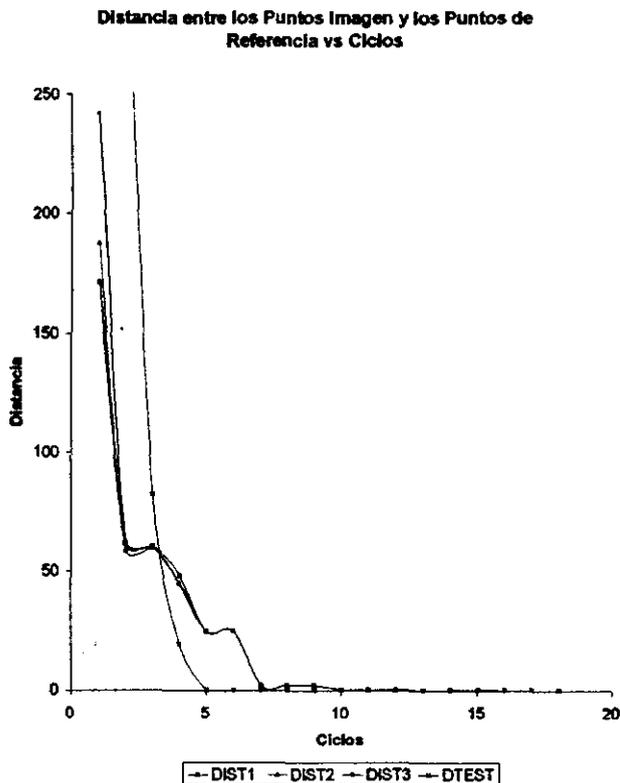


Gráfico 6.8 Distancias entre los puntos imagen y la circunferencia de referencia vs. Ciclos

En este gráfico se puede observar que la curva que representa a las imágenes proyectadas por el láser, tiende a converger en pocos ciclos. La precisión que resulta en esta curva es muy alta, ya que se empleó al láser como condición de comparación.

La distancia de los puntos imagen con respecto a los puntos de referencia es aceptable, logrando una precisión de 0.022 píxeles en la suma de cuadrados. La base de datos que se empleó para hacer esta gráfica está en el Apéndice B (III - 2).

El Gráfico 6.9, que presenta la longitud de los actuadores en función de cada ciclo, es la siguiente:

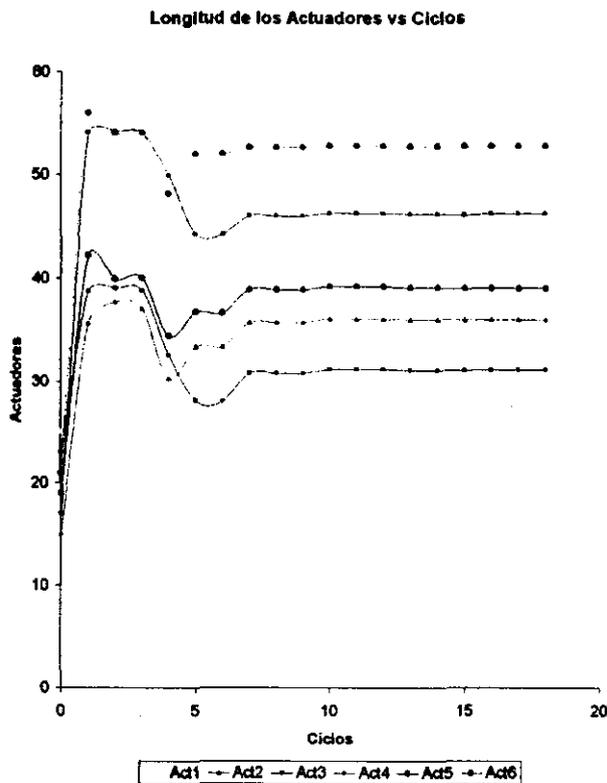


Gráfico 6.9 Longitudes de los Actuadores vs. Ciclos

Este gráfico, al compararlo con el obtenido en la propuesta del Triángulo de Referencia (Gráfico 6.3), permite comprobar que en ambas propuestas se obtienen los mismos resultados.

La base de datos que se empleó para hacer esta gráfica se muestra en el Apéndice B (III - 3).

## 7. CONCLUSIONES

*El proyecto se desarrolló en varios temas, por lo que será necesario describir una conclusión para cada uno de ellos.*

### **Cinemática**

*Tanto el análisis de la Cinemática Directa como el de la Inversa resultaron ser muy eficientes. Los programas además de proporcionar una alta precisión, las siete cifras significativas que permite Basic, son bastante rápidos.*

*El desarrollo de la Cinemática Directa requiere de una sola iteración, la cual, mediante un método numérico adecuado, resulta bastante sencilla. Para lograr una mayor eficiencia y simplificar los cálculos en el control geométrico se aplicaron algunos análisis matemáticos, como la obtención del punto correspondiente de los vértices 2 y 3 en función del vértice 1, ya que a partir de un análisis de las distancias máxima y mínima entre el vértice y las circunferencias 2 y 3, se pueden calcular directamente los puntos correspondientes. Las comprobaciones de las tres matrices numéricas permiten remarcar que el método analítico que se desarrolló es de gran confiabilidad.*

*La Cinemática Inversa se obtuvo de manera muy sencilla y ordenada gracias a que se aplicó el álgebra matricial. La comprobación de las dos cinemáticas resultó ser bastante confiable, puesto que alcanza la precisión que permite Basic. Si se requiere un sistema con mayor precisión se puede sacrificar la rapidez del programa por la inclusión de 14 cifras significativas.*

Gracias a la obtención de las dos cinemáticas, la generación de las trayectorias resulta muy sencilla. La función matemática que define a la trayectoria se puede dar en función de las necesidades que se requieren. La función de interpolación que se presentó para generar trayectorias, resulta ser representativa de la gran ventaja de haber resuelto ambas cinemáticas, ya que en el espacio de trabajo del hexápodo las trayectorias que se emplean son fundamentalmente no-lineales, pero con sólo incluir una función, que es meramente lineal, para desplazar a los actuadores, se puede lograr todo tipo de trayectoria, incluyendo las no-lineales.

### **Dinámica**

La Dinámica se determinó mediante el método de desplazamientos virtuales. Este método presenta dos limitaciones: la primera se da al realizar el desplazamiento de un actuador, provocando que la plataforma móvil también sufra un desplazamiento, pero a diferencia del que tuvo el actuador, éste es no-lineal. Al ser un desplazamiento no-lineal los resultados pueden caer en ciertas incongruencias. Por ejemplo, a diferentes desplazamientos de un actuador, las variaciones que sufre la posición de la plataforma no cambiarán con la misma relación de los desplazamientos del actuador. Por lo que, la máxima fuerza que debe ejercer un actuador como oposición a la fuerza de gravedad, a lo largo de una trayectoria, se puede obtener mediante una tabulación de diversos desplazamientos. Obtener la máxima fuerza mediante derivadas resulta complicado, debido a que la función, que determina el desplazamiento que sufrió la plataforma móvil, es el programa de la Cinemática Directa, donde se emplean iteraciones.

El inconveniente de la no-linearidad se puede resolver si se consideran desplazamientos del actuador tan pequeños, tal que la curva que realiza la trayectoria de la plataforma móvil se puede considerar como lineal. Esto conlleva a la segunda limitación, ya que el programa maneja siete cifras significativas, y la precisión se pierde en el momento que se hace la sustracción entre el estado inicial y el estado final de la posición de la plataforma móvil, para obtener así el incremento que se generó. Este problema puede resolverse si en el programa se utilizan más cifras significativas, pero reduciría la velocidad en los cálculos.

## **Prueba óptica**

*La propuesta que se sugiere en la sección 5.3, para controlar el hexápodo y ser el manipulador que alinee a los dos espejos de un telescopio, presenta algunos puntos que es importante mencionar, tanto en sus ventajas como en sus limitaciones:*

### **Ventajas**

- *La propuesta resulta sencilla y además económica, ya que para lograr la alineación de los espejos sólo se incluyen tres leds y una cámara de video.*
- *La propuesta tiene la gran ventaja de poder controlar los 6 DOF, con sólo la información que presentan las imágenes de los leds en un plano, el plano imagen de la cámara de video.*
- *La propuesta puede considerarse para otras aplicaciones, ya que el sistema no forma parte del telescopio. El criterio que se puede manejar es el de ser un sistema que permite controlar 6 DOF en forma automática.*
- *La propuesta puede ser la base para un sistema más complejo, incluyendo un sistema análogo que permita controlar el espejo secundario del telescopio.*

### **Limitaciones**

- *La propuesta no puede aplicarse al sistema del telescopio, ya que la luz emitida por los leds puede generar aberraciones ópticas. Pero como propuesta base para un sistema más complejo puede resultar muy tentador.*

### **Propuestas de control.**

*En cuanto al control, en las tres propuestas, aparece una contrariedad. El algoritmo primario, "el Desplazamiento de los Vértices", al realizar el movimiento y alcanzar el punto final, o sea satisfacer la condición del algoritmo (la mínima distancia para las propuestas del Triángulo de Referencia con y sin la prueba del láser y la equidistancia para la propuesta de la Circunferencia de Referencia), debe registrarse y regresar a su estado inicial. Este filtro fue necesario para lograr la convergencia del control. Para*

la simulación no presenta ningún inconveniente, ya que se registra y automáticamente se cambia el valor por su estado inicial, pero para una aplicación real resulta una pérdida de tiempo alcanzar un estado mediante el desplazamiento de los actuadores y posteriormente que éstos regresen a su estado anterior.

### Consideraciones en los programas de computadora.

Las matrices de transformación que se emplearon presentan la siguiente configuración:

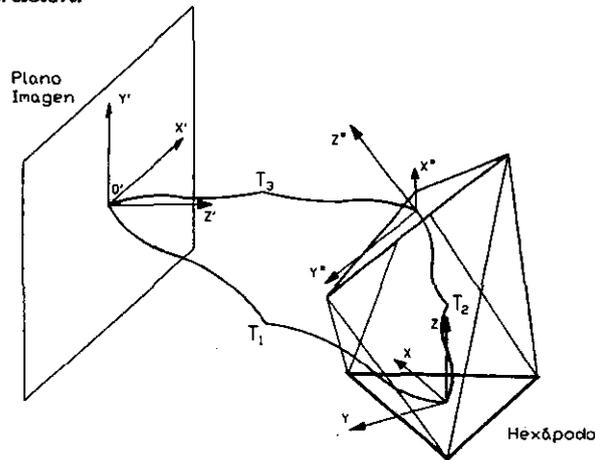


Figura 7.1 Configuración de las matrices de transformación

La función de cada matriz se representa en la siguiente tabla.

$T_1$	Define a la estructura del telescopio.
$T_2$	Define a la matriz de transformación del hexápodo.
$T_3$	Permite expresar las proyecciones de los leds.

La simulación se logra mediante la matriz  $T_3$ , debido a que esta proporciona las proyecciones de los leds. En cambio el control se determina mediante la matriz  $T_2$ , esto significa que cada algoritmo debe referirse mediante esta matriz. Por conveniencia de la programación se realizó la siguiente consideración: en vez de referir los algoritmos a la matriz  $T_2$ , se refirieron a la sustracción entre  $T_1$  y  $T_3$ . Análíticamente no afecta esta consideración a la teoría expuesta en el capítulo 5, ya que es un análisis equivalente. Para una aplicación real tampoco presentaría un problema, porque si se desconoce la información que proporcionaría la estructura del telescopio, debido a las flexiones, el error se corregirá en la matriz  $T_3$ , donde las proyecciones de los leds cambiarán de posición.

Otra consideración muy importante que se tomó en cuenta, misma que se aplica en el Control, es la siguiente: En una aplicación real se controlan directamente las longitudes de los actuadores, pero en este proyecto cada uno de los algoritmos se refiere a la posición de los vértices del triángulo móvil, se aplica la Cinemática Inversa y se obtienen las longitudes de los actuadores. A diferencia que en una aplicación real, los parámetros que se controlan son las longitudes de los actuadores y para conocer la posición de la plataforma móvil se debe aplicar la Cinemática Directa. Debido a que la Cinemática Directa incluye una iteración y representa mayor tiempo de programación, se optó por utilizar la Cinemática Inversa. El utilizar en la simulación, como parámetros, a las posiciones de los vértices del triángulo móvil, es correcto debido a que se conocen ambas cinemáticas.

### **Resultados del Control**

Sobre las dos propuestas del algoritmo primario que se desarrollaron, los mejores resultados se obtuvieron con la condición de mínima distancia, ya que en el caso de la equidistancia se pueden presentar limitaciones que provocan la divergencia. La principal se da cuando el triángulo tiende a ser equilátero o isósceles, ya que el desplazamiento de los vértices proyectará una curva cuasirecta y radial, lo cual provoca que cualquier valor que tome el ángulo  $\theta$ , satisfaga la condición. Esto se resuelve manteniendo en todos los ciclos los algoritmos de Enfoque y Acercamiento, pero son tantas las condiciones que no permitirá al algoritmo primario aproximarse a un valor más preciso.

La propuesta del Triángulo de Referencia con Láser tiene el inconveniente de volver al algoritmo primario un sistema de lazo cerrado, esto provoca que el programa de control tenga tantas condiciones que el desarrollo tiende a caer en ciclos repetitivos, generando así una divergencia.

La propuesta de la Circunferencia de Referencia tiene la desventaja de sólo poder controlar cinco grados de libertad. En cuanto a los resultados que proporciona no son la base para realizar una comparación, ya que en el algoritmo primario de esta propuesta se probó la condición de equidistancia, la cual, como ya se explicó, tiene grandes limitaciones.

Indudablemente la propuesta con que se obtuvieron los mejores resultados es la del Triángulo de Referencia, ya que además de ser la más precisa, controla los seis grados de libertad y resulta ser muy económica, ya que no requiere de un equipo extra como el láser.

## APÉNDICE A (Cinemática Inversa)

Cuando se resuelve un algoritmo, ya sean los secundarios o el primario, se obtienen los nuevos valores de los vértices del triángulo móvil. Con estos valores se puede aplicar la Cinemática Inversa, con el propósito de encontrar las nuevas longitudes de los actuadores. El desarrollo matemático está dado por:

$$L_i = \sqrt{\left({}^{nf}P_{nm}x\right)^2 + \left({}^{nf}P_{nm}y\right)^2 + \left({}^{nf}P_{nm}z\right)^2}$$

Longitud del Actuador $i$	$nf$	$nm$
1	6	2
2	4	2
3	4	1
4	5	1
5	5	3
6	6	3

Expresado de otra forma:

$$L1 = \sqrt{\left({}^6P_2x\right)^2 + \left({}^6P_2y\right)^2 + \left({}^6P_2z\right)^2}$$

$$L4 = \sqrt{\left({}^5P_1x\right)^2 + \left({}^5P_1y\right)^2 + \left({}^5P_1z\right)^2}$$

$$L2 = \sqrt{\left({}^4P_2x\right)^2 + \left({}^4P_2y\right)^2 + \left({}^4P_2z\right)^2}$$

$$L5 = \sqrt{\left({}^5P_3x\right)^2 + \left({}^5P_3y\right)^2 + \left({}^5P_3z\right)^2}$$

$$L3 = \sqrt{\left({}^4P_1x\right)^2 + \left({}^4P_1y\right)^2 + \left({}^4P_1z\right)^2}$$

$$L6 = \sqrt{\left({}^6P_3x\right)^2 + \left({}^6P_3y\right)^2 + \left({}^6P_3z\right)^2}$$

## APÉNDICE B *(Bases de Datos)*

### **Apéndice B (I-1) Lados del triángulo imagen vs Ciclos (Triángulo de Referencia)**

<b>Ciclos</b>	<b>L1</b>	<b>L2</b>	<b>L3</b>
1	135.4139	119.2595	129.9437
2	100.9435	101.5885	106.9442
3	104.7434	103.3508	105.4403
4	154.3535	149.3794	153.7988
5	156.0298	176.4899	152.8665
6	165.6941	175.477	164.9838
7	192.0262	199.0061	192.5168
8	200.0849	199.9485	201.1749
9	195.8959	195.864	196.5828
10	191.401	191.3995	191.9084
11	192.1466	192.1352	192.5474
12	192.0387	192.0343	192.3359
13	191.9124	191.9108	192.1366
14	191.9385	191.9355	192.1067
15	191.9553	191.9535	192.0814
16	191.9693	191.9681	192.0637
17	191.9767	191.9758	192.0474
18	191.9825	191.9818	192.0355
19	191.9869	191.9865	192.0266
20	191.99	191.9898	192.0197
21	191.9926	191.9924	192.0149
22	191.9944	191.9944	192.0112
23	191.9958	191.9957	192.0083
24	191.9968	191.9967	192.0062
25	191.9978	191.9976	192.0047
26	191.9983	191.9982	192.0034
27	191.9987	191.9986	192.0025
28	191.9991	191.9989	192.0019
29	191.9993	191.9992	192.0014
30	191.9995	191.9994	192.0011
31	191.9996	191.9996	192.0008
32	191.9997	191.9997	192.0006
33	191.9998	191.9998	192.0005
34	191.9998	191.9998	192.0003
35	191.9999	191.9998	192.0002
36	191.9999	191.9999	192.0002
37	191.9999	191.9999	192.0001
38	192	191.9999	192.0001
39	192	191.9999	192
40	192	192	192
41	192	192	192

**Apéndice B (I-2) Distancias entre los Puntos Imagen y los  
Puntos de Referencia vs Ciclos  
(Triángulo de Referencia)**

<b>Ciclos</b>	<b>DIST1</b>	<b>DIST2</b>	<b>DIST3</b>	<b>DTEST</b>
1	171.1503	187.5047	242.1627	922.6531
2	60.87025	58.39077	62.18689	1047.531
3	59.32559	58.10587	58.83686	1275.34
4	27.75096	56.27961	47.38777	705.9454
5	16.14235	26.55197	14.11647	827.6885
6	12.32671	17.42501	11.6968	569.3683
7	3.477242	1.343025	3.86245	103.5969
8	5.05865	5.145601	4.374149	163.2998
9	2.513119	2.529855	2.104509	93.18425
10	0.1904736	0.1800386	0.4908393	6.85232
11	0.271787	0.2756303	4.360205E-03	16.97183
12	0.1691682	0.1687389	3.661183E-02	11.50934
13	8.366567E-02	8.170591E-02	9.497777E-02	6.519038
14	6.358287E-02	6.347687E-02	6.902730E-02	5.085238
15	4.803546E-02	4.788251E-02	5.078753E-02	3.977094
16	3.674784E-02	3.658421E-02	3.634983E-02	3.169828
17	2.740694E-02	2.740613E-02	2.742854E-02	2.508211
18	2.050955E-02	2.049397E-02	2.056657E-02	2.017939
19	1.537391E-02	1.537241E-02	1.537202E-02	1.65066
20	1.143018E-03	1.143083E-03	1.157910E-03	1.381434
21	8.594852E-03	8.599902E-03	8.646448E-03	1.183526
22	6.462138E-03	6.475447E-03	6.473708E-03	1.037181
23	4.822381E-03	4.819316E-03	4.887705E-03	0.923554
24	3.598845E-03	3.590417E-03	3.680353E-03	0.8426678
25	2.716450E-03	2.708654E-03	2.687724E-03	0.7814119
26	2.003893E-03	2.005198E-03	2.029451E-03	0.7359961
27	1.489687E-03	1.485174E-03	1.522289E-03	0.7026636
28	1.111754E-03	1.111995E-03	1.146670E-03	0.6778572
29	8.262321E-04	8.246510E-04	8.684534E-04	0.6605963
30	6.310875E-04	6.229503E-04	6.467121E-04	0.6493298
31	4.701992E-04	4.674121E-04	4.743741E-04	0.6382843
32	3.586627E-04	3.596104E-04	3.343904E-04	0.6328602
33	2.674644E-04	2.628663E-04	2.576263E-04	0.6257004
34	1.946991E-04	1.926924E-04	1.078959E-05	0.6237655
35	1.369577E-04	1.266908E-04	1.433433E-04	0.6203054
36	9.658042E-05	9.203493E-05	1.145015E-04	0.6181266
37	6.908711E-05	6.816331E-05	7.854943E-05	0.6168557
38	5.772684E-05	6.409535E-05	5.554283E-05	0.6150206
39	3.814697E-05	3.747004E-05	5.394797E-05	0.6128917
40	2.412626E-05	3.520711E-05	3.814697E-05	0.6128626
41	2.157919E-05	1.631619E-05	1.144409E-05	0.6119395

**Apéndice B (I-3) Longitudes de los Actuadores vs Ciclos  
(Triángulo de Referencia)**

<b>Ciclos</b>	<b>Act1</b>	<b>Act2</b>	<b>Act3</b>	<b>Act4</b>	<b>Act5</b>	<b>Act6</b>
0	25	15	17	23	19	21
1	35.82062	35.64997	54.14884	38.72638	42.20336	56.0864
2	34.73864	35.16668	54.5803	39.67664	36.70385	51.54739
3	36.53648	35.84972	52.00144	35.99033	36.27155	51.67372
4	28.07284	28.86525	47.86145	32.78919	32.4949	46.56205
5	27.01447	34.78194	44.49187	28.76307	37.30326	51.55825
6	30.18542	37.53714	45.54423	30.18166	38.1499	51.86729
7	30.01577	37.40723	46.97167	31.86542	39.3345	52.99509
8	28.78203	36.18883	46.3507	31.24335	39.16768	52.86694
9	28.7579	36.16977	46.34743	31.23631	39.16795	52.87203
10	28.73229	36.14354	46.33309	31.22167	39.16854	52.87177
11	28.65465	36.07022	46.29854	31.18415	39.16465	52.86079
12	28.64914	36.06335	46.29358	31.17979	39.16542	52.87982
13	28.64476	36.05774	46.28898	31.17563	39.16588	52.87869
14	28.64122	36.05333	46.28573	31.17275	39.16627	52.878
15	28.63874	36.05015	46.28321	31.17044	39.16652	52.87739
16	28.63687	36.04778	46.28139	31.1688	39.16672	52.87693
17	28.63549	36.04602	46.28001	31.16754	39.16687	52.87661
18	28.63439	36.04463	46.27898	31.16662	39.16698	52.87635
19	28.63359	36.04362	46.27819	31.16589	39.16707	52.87616
20	28.63307	36.04293	46.27764	31.16541	39.16713	52.87601
21	28.63261	36.04235	46.2772	31.16502	39.16717	52.87591
22	28.63228	36.04193	46.27685	31.16468	39.16721	52.87583
23	28.63204	36.04163	46.27661	31.16446	39.16723	52.87577
24	28.63186	36.04139	46.27643	31.1643	39.16726	52.87573
25	28.63173	36.04122	46.27628	31.16417	39.16727	52.87569
26	28.63161	36.04107	46.27619	31.16408	39.16728	52.87567
27	28.63154	36.04098	46.27612	31.16402	39.16729	52.87565
28	28.63149	36.04091	46.27606	31.16397	39.16729	52.87564
29	28.63146	36.04087	46.27603	31.16395	39.1673	52.87563
30	28.63141	36.04082	46.27599	31.16391	39.1673	52.87562
31	28.63139	36.04079	46.27596	31.16389	39.1673	52.87561
32	28.63137	36.04076	46.27595	31.16387	39.1673	52.87561
33	28.63136	36.04074	46.27594	31.16386	39.16731	52.8756
34	28.63134	36.04073	46.27592	31.16384	39.16731	52.8756
35	28.63134	36.04072	46.27592	31.16384	39.16731	52.8756
36	28.63133	36.04071	46.27591	31.16384	39.16731	52.8756
37	28.63133	36.04071	46.27591	31.16384	39.16731	52.8756
38	28.63132	36.0407	46.2759	31.16383	39.16731	52.8756
39	28.63133	36.0407	46.2759	31.16383	39.16731	52.8756
40	28.63132	36.0407	46.2759	31.16382	39.16731	52.8756
41	28.63132	36.0407	46.2759	31.16383	39.16731	52.8756

**Apéndice B (II-1) Lados del triángulo imagen vs Ciclos  
(Circunferencia de Referencia)**

<b>Ciclos</b>	<b>L1</b>	<b>L2</b>	<b>L3</b>
1	135.4139	119.2595	129.9437
2	173.9448	148.6895	221.1247
3	197.1367	168.1406	205.4193
4	192.8509	189.9305	193.1696
5	192.0522	191.1303	192.8065
6	191.9244	191.6372	192.4358
7	191.8615	191.8992	192.2386
8	191.8228	192.032	192.1448
9	191.7918	192.0984	192.1093
10	191.783	192.1153	192.1011
11	191.7821	192.1165	192.1009
12	191.7791	192.1195	192.1008
13	191.7836	192.1141	192.1018
14	191.7794	192.1183	192.1017
15	191.7865	192.1107	192.1023
16	191.7841	192.1181	192.0972
17	191.7788	192.1203	192.1003
18	191.7799	192.1179	192.1016

**Apéndice B (II-2) Distancias entre los Puntos Imagen y los  
Puntos de Referencia vs Ciclos  
(Circunferencia de Referencia)**

<b>Ciclos</b>	<b>DIST1</b>	<b>DIST2</b>	<b>DIST3</b>	<b>DTEST</b>
1	74.0539	163.3836	40.0991	---
2	3.606170E-06	3.030176E-06	9.619442E-06	922.6481
3	1.551953E-06	4.506918E-06	4.714055E-07	600.7277
4	1.229240E-06	3.955801E-07	7.852055E-07	678.3456
5	7.419054E-07	1.982817E-06	3.438055E-06	695.6061
6	4.321365E-08	4.376575E-07	1.592975E-07	701.6827
7	4.799521E-07	2.835850E-06	1.565054E-06	705.8774
8	1.740119E-06	4.002497E-07	3.484708E-06	709.4059
9	1.064548E-06	2.018879E-06	1.278539E-06	713.3378
10	1.345000E-06	2.405452E-06	8.577409E-07	714.2882
11	1.993728E-06	4.443482E-06	1.983360E-06	714.3187
12	1.104243E-06	1.510809E-06	4.666859E-07	715.1536
13	2.246838E-06	4.039744E-06	2.127890E-06	714.3882
14	1.991831E-06	1.417750E-06	1.487481E-06	715.3459
15	1.830134E-06	6.166787E-06	3.741311E-06	713.5649
16	8.898588E-07	3.840282E-06	1.903123E-06	713.749
17	1.712042E-06	3.634823E-06	2.707186E-06	715.4504
18	2.346952E-06	1.423121E-07	2.298008E-08	715.2206

**Apéndice B (II-3) Longitudes de los Actuadores vs Ciclos  
(Circunferencia de Referencia)**

<b>Ciclos</b>	<b>Act1</b>	<b>Act2</b>	<b>Act3</b>	<b>Act4</b>	<b>Act5</b>	<b>Act6</b>
0	25	15	17	23	19	21
1	24.91777	33.65563	57.45821	39.53679	40.80707	54.70565
2	28.51154	32.73805	54.18096	36.99456	37.53891	50.45368
3	28.29502	31.82366	51.97182	35.41125	36.42463	48.61637
4	27.92563	31.59724	51.66182	35.17508	36.46346	48.64429
5	27.78702	31.50891	51.4965	35.05099	36.50392	48.68608
6	27.72284	31.46559	51.41662	34.98083	36.52069	48.70163
7	27.69565	31.44981	51.37763	34.96161	36.52803	48.70821
8	27.6879	31.4489	51.3606	34.94893	36.52974	48.70958
9	27.68719	31.45037	51.35624	34.94574	36.5302	48.71
10	27.68741	31.45092	51.35582	34.9454	36.5303	48.71009
11	27.68832	31.45222	51.35555	34.94521	36.52994	48.7098
12	27.6881	31.4513	51.35649	34.9459	36.53016	48.71003
13	27.68822	31.45178	51.3561	34.94561	36.52969	48.70956
14	27.68789	31.45092	51.35676	34.94609	36.53058	48.71044
15	27.68816	31.45199	51.3549	34.94487	36.53083	48.71071
16	27.68694	31.45018	51.35589	34.94544	36.52953	48.70931
17	27.68745	31.45068	51.35633	34.94582	36.52963	48.70945
18	27.68722	31.45024	51.35666	34.94603	36.52962	48.70942

**Apéndice B (III-1) Lados del triángulo imagen vs Ciclos  
(Triángulo de Referencia con láser)**

<b>Ciclos</b>	<b>L1</b>	<b>L2</b>	<b>L3</b>
1	135.4139	119.2595	129.9437
2	100.9435	101.5885	106.9442
3	104.7204	103.9587	104.9358
4	149.339	148.8756	149.3188
5	149.8019	149.8389	149.6794
6	149.7827	149.8219	149.715
7	188.6117	188.6612	188.5309
8	188.641	188.6015	188.5622
9	188.6411	188.6015	188.5622
10	192.6608	192.6204	192.5802
11	192.6608	192.6204	192.5802
12	192.6607	192.6204	192.5801
13	191.9142	191.8741	191.8337
14	191.4142	191.874	191.8338
15	191.9141	191.874	191.8338
16	192.0653	192.0251	191.985
17	192.0653	192.0251	191.9849
18	192.052	191.995	191.9548

**Apéndice B (III-2) Distancias entre los Puntos Imagen y los Puntos de Referencia vs Ciclos (Triángulo de Referencia con láser)**

Ciclos	DIST1	DIST2	DIST3	DTEST
1	171.1503	187.5047	242.1627	922.6531
2	60.87025	58.39077	62.18689	339.44470
3	60.20872	59.61781	59.88298	82.03766
4	47.90081	45.14828	44.25896	19.04630
5	24.39610	24.41738	24.32524	0.184778
6	24.38517	24.40778	24.34606	0.04294238
7	1.96748	1.996094	1.924358	0.02159529
8	1.98494	1.96222	1.93939	0.00386415
9	1.984919	1.962205	1.939369	0.00425612
10	0.3352194	0.3592596	0.3817111	0.00411336
11	0.3352038	0.3592288	0.3817050	0.00109513
12	0.3351982	0.3592070	0.3817139	0.01495659
13	0.09691603	0.07759187	0.05126050	0.00254301
14	0.09687614	0.07751573	0.05131603	0.00164233
15	0.09688083	0.07755313	0.05134697	0.00294080
16	0.01595489	0.03045497	0.04000430	0.00309511
17	0.01597212	0.03045222	0.03999236	0.00309199
18	0.02931006	0.02692869	0.02431028	0.00255557

**Apéndice B (III-3) Longitudes de los Actuadores vs Ciclos (Triángulo de Referencia con láser)**

Ciclos	Act1	Act2	Act3	Act4	Act5	Act6
0	25	15	17	23	19	21
1	35.82062	35.64997	54.14884	38.72638	42.20336	56.0864
2	38.82405	37.67846	54.29444	39.04585	39.90903	54.13129
3	37.845	37.04979	54.07321	38.80015	40.05262	54.16447
4	27.68622	30.27124	49.88116	32.56509	34.46764	48.18074
5	27.21608	33.41716	44.24879	28.04907	36.75016	52.12649
6	27.21326	33.40972	44.24632	28.0472	36.748	52.12343
7	28.39682	35.76104	46.04765	30.82869	38.91214	52.75075
8	28.39197	35.75658	46.0539	30.83275	38.91096	52.74996
9	28.39196	35.75657	46.05389	30.83274	38.91095	52.74995
10	28.67499	36.08985	46.31311	31.22004	39.21856	52.90282
11	28.67498	36.08984	46.31311	31.22004	39.21857	52.90283
12	28.67498	36.08984	46.31311	31.22003	39.21862	52.90285
13	28.62142	36.02756	46.26459	31.14776	39.16103	52.87379
14	28.6214	36.02753	46.26459	31.14775	39.16104	52.8738
15	28.62141	36.02755	46.26459	31.14775	39.16103	52.87379
16	28.63223	36.04015	46.2744	31.16237	39.17267	52.87966
17	28.63222	36.04015	46.27441	31.16238	39.17267	52.87965
18	28.63007	36.03764	46.27245	31.15946	39.17035	52.87849

# APÉNDICE C (Programas)

## Cinemática Directa

```

DECLARE SUB A.ENTR (LM, LF, L(), P4(), P5(), P6(), fi1, fi2, fi3)
DECLARE SUB B.TRIG (LF, L(), V1(), V2(), V3(), r1, r2, r3)
DECLARE SUB C.CICLO (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, m, k, LF, LM, r1, r2, r3,
    fi1, fi2, fi3, hn1, L12min, L12max, L13min, L13max, LM(), V1(), V2(),
    V3(), P1(), P2(), P3(), P4(), P5(), P6(), Pd1())
DECLARE SUB C1.Vert1 (Theta1, r1, fi1, hn1, V1(), P1(), Pd1())
DECLARE SUB C2.IT13 (Theta1, Theta3a, Theta3b, LM, r1, r3, fi1, fi3, hn1, L13min, L13max,
    V1(), V3(), P5(), P6(), P1(), P2(), P3(), Pd1())
DECLARE SUB C3.IT12 (Theta1, Theta2a, Theta2b, LM, r1, r2, fi1, fi2, hn1, L12min, L12max,
    L13min, L13max, V1(), V2(), P4(), P6(), P1(), P2(), P3(), Pd1())
DECLARE SUB C4.Conv23 (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, LM(), L12min,
    L12max, L13min, L13max, P2(), P3())
DECLARE SUB C5.POSIC (LF, k, P1(), P2(), P3(), P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z)
DECLARE SUB C51.VECT (rx, ry, rz, P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z)
DECLARE SUB C52.MATRIZ (rx, ry, rz, P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z, R(), RM())
DECLARE SUB C52a.MTRZ.EULER (R(), RM())
DECLARE SUB C52b.MTRZ.VECT.AUX (R())

```

```

CLS
DIM L(10)
DIM V1(1000)
DIM V2(1000)
DIM V3(1000)
DIM P4(1000)
DIM P5(1000)
DIM P6(1000)
DIM P1(1000)
DIM P2(1000)
DIM P3(1000)
DIM Pd1(1000)
DIM LM(100, 100)

```

```

CALL A.ENTR(LM, LF, L(), P4(), P5(), P6(), fi1, fi2, fi3)
CALL B.TRIG(LF, L(), V1(), V2(), V3(), r1, r2, r3)
CALL C.CICLO(Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LF, LM, m, k, r1, r2, r3, fi1, fi2,
    fi3, hn1, L12min, L12max, L13min, L13max, LM(), V1(), V2(), V3(), P1(), P2(),
    P3(), P4(), P5(), P6(), Pd1())

```

END

SUB A.ENTR (LM, LF, L(), P4(), P5(), P6(), fi1, fi2, fi3) STATIC

```

'Entradas
INPUT "De que tamaño es cada lado del triángulo móvil"; LM
INPUT "De que tamaño es cada lado del triángulo fijo"; LF
FOR k = 1 TO 6
    PRINT "Dar el la longitud del actuador"; k; : INPUT L(k)
NEXT k

```

```

'Posición de los vértices del triángulo fijo
P4(1) = LF / (2 * SQR(3)); P4(2) = -LF / 2 : P4(3) = 0
P5(1) = -LF / SQR(3); P5(2) = 0 : P5(3) = 0
P6(1) = LF / (2 * SQR(3)); P6(2) = LF / 2 : P6(3) = 0

```

```

'Cálculo de los ángulos fi
Pi = 3.1415926536#
fi1 = 60 * Pi / 180
fi2 = 180 * Pi / 180
fi3 = 300 * Pi / 180

```

END SUB

```
SUB B.TRIG (LF, L(), V1(), V2(), V3(), r1, r2, r3) STATIC
```

```
'Obtención de los vectores Vi
```

```
Calfa3 = (LF ^ 2 + L(5) ^ 2 - L(6) ^ 2) / (2 * LF * L(5))
```

```
Alfa3 = ATN(SQR(1 - (Calfa3) ^ 2) / Calfa3)
```

```
h3 = L(5) * SIN(Alfa3)
```

```
r3 = h3
```

```
V3(1) = SQR(3) / 2 * (L(5) * Calfa3) - LF / SQR(3)
```

```
V3(2) = .5 * L(5) * Calfa3
```

```
Calfa2 = (LF ^ 2 + L(1) ^ 2 - L(2) ^ 2) / (2 * LF * L(1))
```

```
Alfa2 = ATN(SQR(1 - (Calfa2) ^ 2) / Calfa2)
```

```
h2 = L(1) * SIN(Alfa2)
```

```
r2 = h2
```

```
V2(1) = LF / (2 * SQR(3))
```

```
V2(2) = LF / 2 - L(1) * Calfa2
```

```
Calfa1 = (LF ^ 2 + L(4) ^ 2 - L(3) ^ 2) / (2 * LF * L(4))
```

```
Alfa1 = ATN(SQR(1 - (Calfa1) ^ 2) / Calfa1)
```

```
h1 = L(4) * SIN(Alfa1)
```

```
r1 = h1
```

```
V1(1) = (SQR(3) / 2) * L(4) * Calfa1 - LF / SQR(3)
```

```
V1(2) = -.5 * L(4) * Calfa1
```

```
END SUB
```

```
SUB C.CICLO (Thetal, Theta2a, Theta2b, Theta3a, Theta3b, LF, LM, m, k, r1, r2, r3, fi1, fi2,
            fi3, hnl, L12min, L12max, L13min, L13max, LM(), V1(), V2(), V3(), P1(), P2(),
            P3(), P4(), P5(), P6(), Pdl()) STATIC
```

```
'Condiciones iniciales
```

```
Thetal = 0
```

```
Pi = 3.1415926536#
```

```
g = Pi / 18
```

```
n = 1
```

```
m = 1
```

```
'Donde k representa las 4 posibles soluciones entre el vértice 1 y los corres_
```

```
'pendientes puntos 2 y 3, obtenidos en cada una de las circunferencias.
```

```
'Por otro lado hay que determinar la condición inicial de Theta.
```

```
FOR k = 1 TO 4
```

```
  FOR j = Pi TO 0 STEP -g
```

```
    Thetal = j
```

```
    m = m + 1
```

```
    CALL C1.Vert1(Thetal, r1, fi1, hnl, V1(), P1(), Pdl())
```

```
    CALL C2.IT13(Thetal, Theta3a, Theta3b, LM, r1, r3, fi1, fi3, hnl, L13min, L13max, V1(),
```

```
                V3(), P5(), P6(), P1(), P2(), P3(), Pdl())
```

```
    CALL C3.IT12(Thetal, Theta2a, Theta2b, LM, r1, r2, fi1, fi2, hnl, L12min, L12max,
```

```
                L13min, L13max, V1(), V2(), P4(), P6(), P1(), P2(), P3(), Pdl())
```

```
    CALL C4.Conv23(Thetal, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, LM(), L12min,
```

```
                L12max, L13min, L13max, P2(), P3())
```

```
      IF LM(1, m) <> 0 OR LM(2, m) <> 0 OR LM(3, m) <> 0 OR LM(4, m) <> 0 THEN EXIT FOR
```

```
  NEXT j
```

```
    Thetal = Thetal - g          'La condición inicial de Thetal
```

```
FOR j = Thetal TO 0 STEP -g
```

```
  s = g / n
```

```
  Thetal = j + g - s
```

```
  j = Thetal
```

```
  m = m + 1
```

```

CALL C1.Vert1(Thetal, r1, fil, hnl, V1(), P1(), Pd1())
CALL C2.IT13(Thetal, Theta3a, Theta3b, LM, r1, r3, fil, fi3, hnl, L13min, L13max,
            V1(), V3(), P5(), P6(), P1(), P2(), P3(), Pd1())
CALL C3.IT12(Thetal, Theta2a, Theta2b, LM, r1, r2, fil, fi2, hnl, L12min, L12max,
            L13min, L13max, V1(), V2(), P4(), P6(), P1(), P2(), P3(),
            Pd1())
CALL C4.Conv23(Thetal, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, LM(), L12min,
            L12max, L13min, L13max, P2(), P3())

IF LM(k, m) = LM OR LM(k, m) = LM(k, m - 1) THEN

    PRINT
    PRINT
    PRINT "Los valores finales son para k ="; k

    CALL C5.POSIC(LF, k, P1(), P2(), P3(), P1x, P2x, P3x, P1y, P2y, P3y, P1z,
                P2z, P3z)

    EXIT SUB
END IF

IF LM(k, m) > LM AND LM(k, m - 1) < LM THEN
    j = j + s
    m = m - 1
    n = n * 10
END IF

IF LM(k, m) < LM AND LM(k, m - 1) > LM THEN
    j = j + s
    m = m - 1
    n = n * 10
END IF

NEXT j
NEXT k

END SUB

SUB C1.Vert1 (Thetal, r1, fil, hnl, V1(), P1(), Pd1()) STATIC
'Posición del vértice 1 en función de Thetal

P1(1) = V1(1) + r1 * COS(Thetal) * COS(fil)
P1(2) = V1(2) + r1 * COS(Thetal) * SIN(fil)
P1(3) = V1(3) + r1 * SIN(Thetal)

hnl = r1 * SIN(Thetal)

Pd1(1) = V1(1) + r1 * COS(fil) * COS(Thetal)
Pd1(2) = V1(2) + r1 * SIN(fil) * COS(Thetal)

END SUB

```

```
SUB C2.IT13 (Theta1, Theta3a, Theta3b, LM, r1, r3, fi1, fi3, hnl, L13min, L13max, V1(),
V3(), P5(), P6(), P1(), P2(), P3(), Pd1()) STATIC
```

```
DIM P3min(1000)
```

```
DIM P3max(1000)
```

```
DIM m3(1000)
```

```
DIM n3(1000)
```

```
Pi = 3.1415926536#
```

```
'Obtención de los vectores n3 y m3.
```

```
FOR k = 1 TO 3
```

```
    m3(k) = Pd1(k) - P6(k)
```

```
    n3(k) = P5(k) - P6(k)
```

```
NEXT k
```

```
'Producto cruz
```

```
    Cruz1 = m3(1) * n3(2) - m3(2) * n3(1)
```

```
'Determinante
```

```
    Det1 = SQR(n3(1) ^ 2 + n3(2) ^ 2)
```

```
'Distancia más corta
```

```
d1 = ABS(Cruz1) / ABS(Det1)
```

```
'Angulo phi entre el vértice 1 y el 3
```

```
phi13 = ATN(hnl / d1)
```

```
'Punto obtenido por la distancia mínima
```

```
P3min(1) = r3 * COS(phi13) * COS(fi3) + V3(1)
```

```
P3min(2) = r3 * COS(phi13) * SIN(fi3) + V3(2)
```

```
P3min(3) = r3 * SIN(phi13) + V3(3)
```

```
'Punto obtenido por la distancia máxima
```

```
P3max(1) = V3(1) - r3 * COS(phi13) * COS(fi3)
```

```
P3max(2) = V3(2) - r3 * COS(phi13) * SIN(fi3)
```

```
P3max(3) = V3(3) - r3 * SIN(phi13)
```

```
'Magnitud de las distancias, mínima y máxima.
```

```
L13min = SQR((P3min(1) - P1(1)) ^ 2 + (P3min(2) - P1(2)) ^ 2 + (P3min(3) - P1(3)) ^ 2)
```

```
L13max = SQR((P3max(1) - P1(1)) ^ 2 + (P3max(2) - P1(2)) ^ 2 + (P3max(3) - P1(3)) ^ 2)
```

```
'Condiciones de convergencia
```

```
IF L13min > LM OR L13max < LM THEN
```

```
    Theta3a = 0
```

```
    Theta3b = 0
```

```
    EXIT SUB
```

```
END IF
```

```
'Obtención del ángulo Gamma3
```

```
CGamma3 = (L13min ^ 2 + L13max ^ 2 - 2 * LM ^ 2) / (L13max ^ 2 - L13min ^ 2)
```

```
IF CGamma3 > 0 THEN
```

```
    Gamma3 = ATN(SQR(1 - CGamma3 ^ 2) / CGamma3)
```

```
END IF
```

```
IF CGamma3 < 0 THEN
```

```
    Gamma3 = ATN(SQR(1 - CGamma3 ^ 2) / CGamma3)
```

```
    Gamma3 = Gamma3 + Pi
```

```
END IF
```

```
'Obtención de los dos ángulos Theta3
```

```
Theta3a = phi13 + Gamma3
```

```
Theta3b = phi13 - Gamma3
```

```
IF Theta3a < 0 THEN Theta3a = Theta3a + 2 * Pi
```

```
IF Theta3b < 0 THEN Theta3b = Theta3b + 2 * Pi
```

```
'Cálculo de los vectores de posición que definen a los vértices.
```

```
P3(1) = V3(1) + r3 * COS(Theta3a) * COS(fi3)
```

```
P3(2) = V3(2) + r3 * COS(Theta3a) * SIN(fi3)
```

```
P3(3) = V3(3) + r3 * SIN(Theta3a)
```

```
P3(11) = V3(1) + r3 * COS(Theta3b) * COS(fi3)
```

```
P3(22) = V3(2) + r3 * COS(Theta3b) * SIN(fi3)
```

```
P3(33) = V3(3) + r3 * SIN(Theta3b)
```

```
END SUB
```

```

SUB C3.IT12 (Theta1, Theta2a, Theta2b, LM, r1, r2, fi1, fi2, hn1, L12min, L12max, L13min,
L13max, V1(), V2(), P4(), P6(), P1(), P2(), P3(), Pd1()) STATIC

DIM P2min(1000)
DIM P2max(1000)
DIM m2(1000)
DIM n2(1000)
Pi = 3.1415926536#

'Obtención de los vectores n2 y m3.
FOR k = 1 TO 3
    m2(k) = Pd1(k) - P6(k)
    n2(k) = P4(k) - P6(k)
NEXT k

'Producto cruz
Cruz1 = m2(1) * n2(2) - m2(2) * n2(1)
'Determinante
Det1 = SQR(n2(1) ^ 2 + n2(2) ^ 2)

'Distancia más corta
d1 = ABS(Cruz1) / ABS(Det1)

'Angulo phi entre el vértice 1 y el 2
phil2 = ATN(hn1 / d1)

'Punto obtenido por la distancia mínima
P2min(1) = r2 * COS(phil2) * COS(fi2) + V2(1)
P2min(2) = r2 * COS(phil2) * SIN(fi2) + V2(2)
P2min(3) = r2 * SIN(phil2) + V2(3)

'Punto obtenido por la distancia máxima
P2max(1) = V2(1) - r2 * COS(phil2) * COS(fi2)
P2max(2) = V2(2) - r2 * COS(phil2) * SIN(fi2)
P2max(3) = V2(3) - r2 * SIN(phil2)

'Magnitud de las distancias, mínima y máxima
L12min = SQR((P2min(1) - P1(1)) ^ 2 + (P2min(2) - P1(2)) ^ 2 + (P2min(3) - P1(3)) ^ 2)
L12max = SQR((P2max(1) - P1(1)) ^ 2 + (P2max(2) - P1(2)) ^ 2 + (P2max(3) - P1(3)) ^ 2)

'Condiciones de convergencia
IF L12min > LM OR L12max < LM THEN
    Theta2a = 0
    Theta2b = 0
    EXIT SUB
END IF

'Obtención del ángulo Gamma2
CGamma2 = (L12min ^ 2 + L12max ^ 2 - 2 * LM ^ 2) / (L12max ^ 2 - L12min ^ 2)
IF CGamma2 > 0 THEN
    Gamma2 = ATN(SQR(1 - CGamma2 ^ 2) / CGamma2)
END IF
IF CGamma2 < 0 THEN
    Gamma2 = ATN(SQR(1 - CGamma2 ^ 2) / CGamma2)
    Gamma2 = Gamma2 + Pi
END IF

'Obtención de los ángulos Theta2
Theta2a = phil2 + Gamma2
Theta2b = phil2 - Gamma2
IF Theta2a < 0 THEN Theta2a = Theta2a + 2 * Pi
IF Theta2b < 0 THEN Theta2b = Theta2b + 2 * Pi

'Cálculo de los vectores de posición que definen a los vértices
P2(1) = V2(1) - r2 * COS(Theta2a)
P2(2) = V2(2)
P2(3) = V2(3) + r2 * SIN(Theta2a)
P2(11) = V2(1) - r2 * COS(Theta2b) * COS(fi2)
P2(22) = V2(2)
P2(33) = V2(3) + r2 * SIN(Theta2b)
END SUB

```

```
SUB C4.Conv23 (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, LM(), L12min, L12max,
              L13min, L13max, P2(), P3()) STATIC
```

```
Pi = 3.1415926536#
```

```
'Convergencia de los vértices 2 y 3
```

```
LM(1, m) = SQR((P2(1) - P3(1)) ^ 2 + (P2(2) - P3(2)) ^ 2 + (P2(3) - P3(3)) ^ 2)
LM(2, m) = SQR((P2(11) - P3(11)) ^ 2 + (P2(22) - P3(22)) ^ 2 + (P2(33) - P3(33)) ^ 2)
LM(3, m) = SQR((P2(1) - P3(11)) ^ 2 + (P2(2) - P3(22)) ^ 2 + (P2(3) - P3(33)) ^ 2)
LM(4, m) = SQR((P2(11) - P3(11)) ^ 2 + (P2(22) - P3(22)) ^ 2 + (P2(33) - P3(33)) ^ 2)
```

```
'Condiciones de convergencia
```

```
IF Theta2a < 0 OR Theta2a > Pi THEN LM(1, m) = 0
IF Theta2a < 0 OR Theta2a > Pi THEN LM(3, m) = 0
IF Theta2b < 0 OR Theta2b > Pi THEN LM(2, m) = 0
IF Theta2b < 0 OR Theta2b > Pi THEN LM(4, m) = 0
IF Theta3a < 0 OR Theta3a > Pi THEN LM(1, m) = 0
IF Theta3a < 0 OR Theta3a > Pi THEN LM(2, m) = 0
IF Theta3b < 0 OR Theta3b > Pi THEN LM(3, m) = 0
IF Theta3b < 0 OR Theta3b > Pi THEN LM(4, m) = 0
IF L12min > LM OR L12max < LM OR L13min > LM OR L13max < LM THEN
  LM(1, m) = 0:   LM(2, m) = 0:   LM(3, m) = 0:   LM(4, m) = 0
END IF
```

```
'Impresión de la iteración
```

```
SELECT CASE k
```

```
CASE 1
```

```
IF LM(1, m) = 0 THEN
```

```
PRINT
```

```
PRINT " Theta1"; , " Theta2a"; , "Theta3a "
```

```
PRINT " "; Theta1 * 180 / Pi; , Theta2a * 180 / Pi; , Theta3a * 180 / Pi
```

```
END IF
```

```
IF LM(1, m) <> 0 THEN
```

```
PRINT
```

```
PRINT " Theta1"; , " Theta2a"; , "Theta3a "; , " LM"; , "LM1 (P2a-P3a)"
```

```
PRINT " "; Theta1 * 180 / Pi; , Theta2a * 180 / Pi; , Theta3a * 180 / Pi; , " "; LM; , LM(1, m)
```

```
END IF
```

```
CASE 2
```

```
IF LM(2, m) = 0 THEN
```

```
PRINT
```

```
PRINT " Theta1"; , " Theta2a"; , "Theta3b "
```

```
PRINT " "; Theta1 * 180 / Pi; , Theta2a * 180 / Pi; , Theta3b * 180 / Pi
```

```
END IF
```

```
IF LM(2, m) <> 0 THEN
```

```
PRINT
```

```
PRINT " Theta1"; , " Theta2a"; , "Theta3b "; , " LM"; , "LM2 (P2b-P3a)"
```

```
PRINT " "; Theta1 * 180 / Pi; , Theta2a * 180 / Pi; , Theta3b * 180 / Pi; , " "; LM; , LM(2, m)
```

```
END IF
```

```
CASE 3
```

```
IF LM(3, m) = 0 THEN
```

```
PRINT
```

```
PRINT " Theta1"; , " Theta2b"; , "Theta3a "
```

```
PRINT " "; Theta1 * 180 / Pi; , Theta2b * 180 / Pi; , Theta3a * 180 / Pi
```

```
END IF
```

```
IF LM(3, m) <> 0 THEN
```

```
PRINT
```

```
PRINT " Theta1"; , " Theta2b"; , "Theta3a "; , " LM"; , "LM1 (P2a-P3b)"
```

```
PRINT " "; Theta1 * 180 / Pi; , Theta2b * 180 / Pi; , Theta3a * 180 / Pi; , " "; LM; , LM(3, m)
```

```
END IF
```

```
CASE 4
```

```
IF LM(3, m) = 0 THEN
```

```
PRINT
```

```
PRINT " Theta1"; , " Theta2b"; , "Theta3b "
```

```
PRINT " "; Theta1 * 180 / Pi; , Theta2b * 180 / Pi; , Theta3b * 180 / Pi
```

```
END IF
```

```
IF LM(4, m) <> 0 THEN
```

```
PRINT
```

```
PRINT " Theta1"; , " Theta2b"; , "Theta3b "; , " LM"; , "LM4 (P2b-P3b)"
```

```
PRINT " "; Theta1 * 180 / Pi; , Theta2b * 180 / Pi; , Theta3b * 180 / Pi; , " "; LM; , LM(4, m)
```

```
END IF
```

```
END SELECT
```

```
END SUB
```

SUB CS.POSIC (LF, k, P1(), P2(), P3(), P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z) STATIC

DIM R(10, 10)  
DIM RM(10, 10)

'Posición final de los vértices 1, 2 y 3  
P1x = P1(1): P1y = P1(2): P1z = P1(3)

SELECT CASE k

CASE 1

P2x = P2(1): P3x = P3(1)  
P2y = P2(2): P3y = P3(2)  
P2z = P2(3): P3z = P3(3)

CASE 2

P2x = P2(11): P3x = P3(1)  
P2y = P2(22): P3y = P3(2)  
P2z = P2(33): P3z = P3(3)

CASE 3

P2x = P2(1): P3x = P3(11)  
P2y = P2(2): P3y = P3(22)  
P2z = P2(3): P3z = P3(33)

CASE 4

P2x = P2(11): P3x = P3(11)  
P2y = P2(22): P3y = P3(22)  
P2z = P2(33): P3z = P3(33)

END SELECT

'Impresión de la posición final de los vértices 1, 2 y 3

PRINT  
PRINT " P1(x,y,z) ", " P2(x, y, z) ", " P3(x, y, z)"  
PRINT  
PRINT " "; P1x, " "; P2x, " "; P3x  
PRINT " "; P1y, " "; P2y, " "; P3y  
PRINT " "; P1z, " "; P2z, " "; P3z

'Posición de los vértices del triángulo fijo

P4(1) = LF / (2 \* SQR(3))  
P4(2) = -LF / 2  
P4(3) = 0

P5(1) = -LF / SQR(3)  
P5(2) = 0  
P5(3) = 0

P6(1) = LF / (2 \* SQR(3))  
P6(2) = LF / 2  
P6(3) = 0

'Comprobación mediante la Cinemática Directa

L(1) = SQR((P6(1) - P2(1)) ^ 2 + (P6(2) - P2(2)) ^ 2 + (P6(3) - P2(3)) ^ 2)  
L(2) = SQR((P4(1) - P2(1)) ^ 2 + (P4(2) - P2(2)) ^ 2 + (P4(3) - P2(3)) ^ 2)  
L(3) = SQR((P4(1) - P1(1)) ^ 2 + (P4(2) - P1(2)) ^ 2 + (P4(3) - P1(3)) ^ 2)  
L(4) = SQR((P5(1) - P1(1)) ^ 2 + (P5(2) - P1(2)) ^ 2 + (P5(3) - P1(3)) ^ 2)  
L(5) = SQR((P5(1) - P3(1)) ^ 2 + (P5(2) - P3(2)) ^ 2 + (P5(3) - P3(3)) ^ 2)  
L(6) = SQR((P6(1) - P3(1)) ^ 2 + (P6(2) - P3(2)) ^ 2 + (P6(3) - P3(3)) ^ 2)

'Impresión de la longitud de los actuadores

PRINT  
FOR k = 1 TO 6  
PRINT " La longitud del actuador"; k; "es igual a"; L(k)  
NEXT k

'La posición del triángulo móvil se define como, el vector de posición y la  
'matriz de rotación

CALL CS1.VECT(rx, ry, rz, P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z)  
CALL CS2.MATRIZ(rx, ry, rz, P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z, R(), RM())  
END SUB

```

SUB C51.VECT (rx, ry, rz, P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z) STATIC
'Vector de posición
rx = (P1x + P2x + P3x) / 3
ry = (P1y + P2y + P3y) / 3
rz = (P1z + P2z + P3z) / 3

PRINT
PRINT " El vector de posición es"
PRINT
PRINT "      r (": rx; ",": ry; ",": rz; ")"

END SUB

SUB C52.MATRIZ (rx, ry, rz, P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z, R(), RM()) STATIC
'Matriz de rotación
'Para obtener la matriz de rotación utilizo los cosenos directores
'Por otro lado traslado el triángulo móvil al fijo.

V1x = P1x - rx: V1y = P1y - ry: V1z = P1z - rz
V2x = P2x - rx: V2y = P2y - ry: V2z = P2z - rz
V3x = P3x - rx: V3y = P3y - ry: V3z = P3z - rz
V2mod = SQR(V2x ^ 2 + V2y ^ 2 + V2z ^ 2)

'El eje X' con respecto al sistema de referencia inicial
R(1, 1) = V2x / V2mod
R(2, 1) = V2y / V2mod
R(3, 1) = V2z / V2mod

'El eje Y' con respecto al sistema de referencia inicial
C23x = -(2 * V1x + V2x) / 3
C23y = -(2 * V1y + V2y) / 3
C23z = -(2 * V1z + V2z) / 3
C23mod = SQR(C23x ^ 2 + C23y ^ 2 + C23z ^ 2)

R(1, 2) = C23x / C23mod
R(2, 2) = C23y / C23mod
R(3, 2) = C23z / C23mod

'El eje Z' con respecto al sistema de referencia inicial
Zx = V2y * C23z - C23y * V2z
Zy = C23x * V2z - V2x * C23z
Zz = V2x * C23y - C23x * V2y
Zmod = SQR(Zx ^ 2 + Zy ^ 2 + Zz ^ 2)

R(1, 3) = Zx / Zmod
R(2, 3) = Zy / Zmod
R(3, 3) = Zz / Zmod

'Impresión de la matriz de rotación
PRINT
PRINT " La matriz de rotación que se generó es"
FOR i = 1 TO 3
  FOR h = 1 TO 3
    PRINT R(i, h); "      ",
  NEXT h
NEXT i

'Cálculo e impresión de las otras matrices
LOCATE 25, 10: PRINT "ENTER para continuar"
DO
LOOP UNTIL INKEY$ = CHR$(13)
CALL C52a.MTRZ.EULER(R(), RM())

LOCATE 25, 10: PRINT "ENTER para continuar"
DO
LOOP UNTIL INKEY$ = CHR$(13)
CALL C52b.MTRZ.VECT.AUX(R())

END SUB

```

```

SUB C52a.MTRZ.EULER (R(), RM()) STATIC

STheta = -R(3, 1)
Theta = ATN(STheta / SQR(1 - STheta ^ 2))

SPhi = R(2, 1) / COS(Theta)
Phi = ATN(SPhi / SQR(1 - SPhi ^ 2))

SPsi = R(3, 2) / COS(Theta)
Psi = ATN(SPsi / SQR(1 - SPsi ^ 2))

PRINT "
PRINT
PRINT "   Los ángulos de Euler son"
PRINT
PRINT "   Phi(z)", " Theta(y)", " Psi(x)"

Pi = 3.1415926536#
PRINT "   ", Phi * 180 / Pi, Theta * 180 / Pi, Psi * 180 / Pi

RM(1, 1) = COS(Theta) * COS(Phi)
RM(2, 1) = COS(Theta) * SIN(Phi)
RM(3, 1) = -SIN(Theta)
RM(1, 2) = SIN(Theta) * COS(Phi) * SIN(Psi) - SIN(Phi) * COS(Psi)
RM(2, 2) = SIN(Theta) * SIN(Phi) * SIN(Psi) + COS(Phi) * COS(Psi)
RM(3, 2) = COS(Theta) * SIN(Psi)
RM(1, 3) = SIN(Theta) * COS(Phi) * COS(Psi) + SIN(Phi) * SIN(Psi)
RM(2, 3) = SIN(Theta) * SIN(Phi) * COS(Psi) - COS(Phi) * SIN(Psi)
RM(3, 3) = COS(Theta) * COS(Psi)

'Impresión de la matriz de rotación de Euler

PRINT
PRINT "   La matriz de rotación de Euler"
PRINT
FOR i = 1 TO 3
  FOR h = 1 TO 3
    PRINT RM(i, h); "   ",
  NEXT h
NEXT i

END SUB

```

```

SUB C52b.MTRZ.VECT.AUX (R()) STATIC

DIM RV(10, 10)

'El ángulo de rotación se obtiene de la siguiente forma

CALfa = (R(1, 1) + R(2, 2) + R(3, 3) - 1) / 2
Alfa = ATN(SQR(1 - CALfa ^ 2) / CALfa)

'Mientras que las componentes del vector k son

IF Alfa = 0 THEN
  kx = 1: ky = 1: kz = 1
ELSE
  kx = 1 / (2 * SIN(Alfa)) * (R(3, 2) - R(2, 3))
  ky = 1 / (2 * SIN(Alfa)) * (R(1, 3) - R(3, 1))
  kz = 1 / (2 * SIN(Alfa)) * (R(2, 1) - R(1, 2))
END IF

Pi = 3.1415926536#

PRINT "
PRINT
PRINT " El ángulo de rotación es "
PRINT
PRINT " Alfa = ", Alfa * 180 / Pi
PRINT
PRINT " Mientras las componentes del vector auxiliar son"
PRINT
PRINT " k (", kx, ", ", ky, ", ", kz, ")"
PRINT

CALfa = COS(Alfa)
SALfa = SIN(Alfa)
VALfa = 1 - COS(Alfa)

RV(1, 1) = kx ^ 2 * VALfa + CALfa
RV(2, 1) = kx * ky * VALfa + kz * SALfa
RV(3, 1) = kx * kz * VALfa - ky * SALfa
RV(1, 2) = kx * ky * VALfa - kz * SALfa
RV(2, 2) = ky ^ 2 * VALfa + CALfa
RV(3, 2) = ky * kz * VALfa + kx * SALfa
RV(1, 3) = kx * kz * VALfa + ky * SALfa
RV(2, 3) = ky * kz * VALfa - kx * SALfa
RV(3, 3) = kz ^ 2 * VALfa + CALfa

'Impresión de la matriz de rotación con un vector auxiliar

PRINT
PRINT " La matriz de rotación con vector auxiliar"
PRINT
FOR i = 1 TO 3
  FOR h = 1 TO 3
    PRINT RV(i, h); " ",
  NEXT h
NEXT i
PRINT "
END SUB

```

## Cinemática Inversa (Euler)

'Este programa permite obtener las longitudes de los actuadores mediante  
'el método de una matriz de rotación de EULER

```
DECLARE SUB A.ENTR (LM, LF, rx, ry, rz, Psi, Theta, Phi, P1(), P2(), P3(), P4(), P5(), P6())
DECLARE SUB B.MTRZ (rx, ry, rz, Phi, Theta, Psi, RE())
DECLARE SUB C.DESARROLLO (P1(), P2(), P3(), P4(), P5(), P6(), RE(), L())
```

```
CLS
DIM RE(4, 4)
DIM L(6, 4)
DIM P1(4)
DIM P2(4)
DIM P3(4)
DIM P4(4)
DIM P5(4)
DIM P6(4)
```

```
PRINT "Este programa permite obtener las longitudes de los actuadores mediante"
PRINT "el m, todo de una matriz de rotación de EULER"
```

```
CALL A.ENTR(LM, LF, rx, ry, rz, Psi, Theta, Phi, P1(), P2(), P3(), P4(), P5(), P6())
CALL B.MTRZ(rx, ry, rz, Phi, Theta, Psi, RE())
CALL C.DESARROLLO(P1(), P2(), P3(), P4(), P5(), P6(), RE(), L())
```

```
END
```

```
SUB A.ENTR (LM, LF, rx, ry, rz, Psi, Theta, Phi, P1(), P2(), P3(), P4(), P5(), P6()) STATIC
```

```
'Entradas
INPUT "El tamaño de cada lado del triángulo móvil"; LM
INPUT "El tamaño de cada lado del triángulo fijo"; LF
INPUT "Dar el valor las coordenadas del centro del triángulo móvil X, Y, Z"; rx, ry, rz
INPUT "Dar la magnitud del ángulo de desviación (sobre el eje x)"; Psi
INPUT "Dar la magnitud del ángulo de elevación (sobre el eje y)"; Theta
INPUT "Dar la magnitud del ángulo de giro (sobre el eje z)"; Phi
```

```
Pi = 3.14159265359#
Phi = Phi * Pi / 180
Theta = Theta * Pi / 180
Psi = Psi * Pi / 180
```

```
'Posición de los vértices con respecto al centro de cada triángulo
```

```
P1(1) = -LM / (2 * SQR(3))
P1(2) = -LM / 2
P1(3) = 0
```

```
P2(1) = LM / SQR(3)
P2(2) = 0
P2(3) = 0
```

```
P3(1) = -LM / (2 * SQR(3))
P3(2) = LM / 2
P3(3) = 0
```

```
P4(1) = -LF / (2 * SQR(3))
P4(2) = LF / 2
P4(3) = 0
```

```
P5(1) = LF / SQR(3)
P5(2) = 0
P5(3) = 0
```

```
P6(1) = -LF / (2 * SQR(3))
P6(2) = -LF / 2
P6(3) = 0
END SUB
```

SUB B.MTRZ (rx, ry, rz, Phi, Theta, Psi, RE()) STATIC

'Estos valores representan a la matriz de rotación

```
RE(1, 1) = COS(Theta) * COS(Phi)
RE(2, 1) = COS(Theta) * SIN(Phi)
RE(3, 1) = -SIN(Theta)
RE(1, 2) = SIN(Theta) * COS(Phi) * SIN(Psi) - SIN(Phi) * COS(Psi)
RE(2, 2) = SIN(Theta) * SIN(Phi) * SIN(Psi) + COS(Phi) * COS(Psi)
RE(3, 2) = COS(Theta) * SIN(Psi)
RE(1, 3) = SIN(Theta) * COS(Phi) * COS(Psi) + SIN(Phi) * SIN(Psi)
RE(2, 3) = SIN(Theta) * SIN(Phi) * COS(Psi) - COS(Phi) * SIN(Psi)
RE(3, 3) = COS(Theta) * COS(Psi)
```

'Estos valores representan al vector de posición

```
RE(1, 4) = rx
RE(2, 4) = ry
RE(3, 4) = rz
```

'Los valores de la transformación de perspectiva

```
RE(4, 1) = 0
RE(4, 2) = 0
RE(4, 3) = 0
RE(4, 4) = 1
```

'Impresión de la matriz de Euler

```
PRINT
PRINT " La matriz de transformación es:"
PRINT
FOR i = 1 TO 3
  FOR j = 1 TO 4
    PRINT " "; RE(i, j); " ";
  NEXT j
  PRINT
NEXT i
PRINT " "; RE(4, 1), " "; RE(4, 2), " "; RE(4, 3), " "; RE(4, 4)

END SUB
```

SUB C.DESARROLLO (P1(), P2(), P3(), P4(), P5(), P6(), RE(), L()) STATIC

'Obtención de la longitud de los actuadores

```
L(1, 1) = RE(1, 1) * P2(1) + RE(1, 2) * P2(2) + RE(1, 3) * P2(3) + RE(1, 4) * P2(4) + P6(1)
L(1, 2) = RE(2, 1) * P2(1) + RE(2, 2) * P2(2) + RE(2, 3) * P2(3) + RE(2, 4) * P2(4) + P6(2)
L(1, 3) = RE(3, 1) * P2(1) + RE(3, 2) * P2(2) - RE(3, 3) * P2(3) + RE(3, 4) * P2(4) + P6(3)

L(2, 1) = RE(1, 1) * P2(1) + RE(1, 2) * P2(2) - RE(1, 3) * P2(3) + RE(1, 4) * P2(4) + P4(1)
L(2, 2) = RE(2, 1) * P2(1) + RE(2, 2) * P2(2) + RE(2, 3) * P2(3) + RE(2, 4) * P2(4) + P4(2)
L(2, 3) = RE(3, 1) * P2(1) + RE(3, 2) * P2(2) + RE(3, 3) * P2(3) + RE(3, 4) * P2(4) + P4(3)

L(3, 1) = RE(1, 1) * P1(1) + RE(1, 2) * P1(2) + RE(1, 3) * P1(3) + RE(1, 4) * P1(4) + P4(1)
L(3, 2) = RE(2, 1) * P1(1) + RE(2, 2) * P1(2) + RE(2, 3) * P1(3) + RE(2, 4) * P1(4) + P4(2)
L(3, 3) = RE(3, 1) * P1(1) + RE(3, 2) * P1(2) + RE(3, 3) * P1(3) + RE(3, 4) * P1(4) + P4(3)

L(4, 1) = RE(1, 1) * P1(1) + RE(1, 2) * P1(2) + RE(1, 3) * P1(3) + RE(1, 4) * P1(4) + P5(1)
L(4, 2) = RE(2, 1) * P1(1) + RE(2, 2) * P1(2) + RE(2, 3) * P1(3) + RE(2, 4) * P1(4) + P5(2)
L(4, 3) = RE(3, 1) * P1(1) + RE(3, 2) * P1(2) + RE(3, 3) * P1(3) + RE(3, 4) * P1(4) + P5(3)

L(5, 1) = RE(1, 1) * P3(1) + RE(1, 2) * P3(2) + RE(1, 3) * P3(3) + RE(1, 4) * P3(4) + P5(1)
L(5, 2) = RE(2, 1) * P3(1) + RE(2, 2) * P3(2) + RE(2, 3) * P3(3) + RE(2, 4) * P3(4) + P5(2)
L(5, 3) = RE(3, 1) * P3(1) + RE(3, 2) * P3(2) + RE(3, 3) * P3(3) + RE(3, 4) * P3(4) + P5(3)

L(6, 1) = RE(1, 1) * P3(1) + RE(1, 2) * P3(2) + RE(1, 3) * P3(3) + RE(1, 4) * P3(4) + P6(1)
L(6, 2) = RE(2, 1) * P3(1) + RE(2, 2) * P3(2) + RE(2, 3) * P3(3) + RE(2, 4) * P3(4) + P6(2)
L(6, 3) = RE(3, 1) * P3(1) + RE(3, 2) * P3(2) + RE(3, 3) * P3(3) + RE(3, 4) * P3(4) + P6(3)
```

```
PRINT
PRINT
PRINT
```

```

LOCATE 11, 1: PRINT " Vértice 1   Vértice 2   Vértice 3"
FOR i = 1 TO 3
  LOCATE 12 + i, 2: PRINT L(3, i) - P4(i)
  LOCATE 12 + i, 15: PRINT L(1, i) - P6(i)
  LOCATE 12 + i, 28: PRINT L(5, i) - P5(i)
PRINT
NEXT i
PRINT

Ac1 = SQR(L(1, 1) ^ 2 + L(1, 2) ^ 2 + L(1, 3) ^ 2)
Ac2 = SQR(L(2, 1) ^ 2 + L(2, 2) ^ 2 + L(2, 3) ^ 2)
Ac3 = SQR(L(3, 1) ^ 2 + L(3, 2) ^ 2 + L(3, 3) ^ 2)
Ac4 = SQR(L(4, 1) ^ 2 + L(4, 2) ^ 2 + L(4, 3) ^ 2)
Ac5 = SQR(L(5, 1) ^ 2 + L(5, 2) ^ 2 + L(5, 3) ^ 2)
Ac6 = SQR(L(6, 1) ^ 2 + L(6, 2) ^ 2 + L(6, 3) ^ 2)

PRINT "      Longitud del Actuador 1 = "; Ac1
PRINT "      Longitud del Actuador 2 = "; Ac2
PRINT "      Longitud del Actuador 3 = "; Ac3
PRINT "      Longitud del Actuador 4 = "; Ac4
PRINT "      Longitud del Actuador 5 = "; Ac5
PRINT "      Longitud del Actuador 6 = "; Ac6

END SUB

```

## **Cinemática Inversa (Vector Auxiliar)**

'Este programa permite obtener las longitudes de los actuadores mediante el método de una matriz de rotación con respecto a un eje arbitrario o vector auxiliar.

```

DECLARE SUB A.ENTR (LM, LF, rx, ry, rz, kx, ky, kz, Alfa, P1(), P2(), P3(), P4(), P5(),
P6())
DECLARE SUB B.MTRZ (rx, ry, rz, kx, ky, kz, Alfa, RV())
DECLARE SUB C.DESARROLLO (P1(), P2(), P3(), P4(), P5(), P6(), RV(), L())

CLS
DIM RV(4, 4)
DIM L(6, 4)
DIM P1(4)
DIM P2(4)
DIM P3(4)
DIM P4(4)
DIM P5(4)
DIM P6(4)

PRINT " Este programa permite obtener las longitudes de los actuadores mediante el método "
PRINT " de una matriz de rotación con respecto a un eje arbitrario o vector auxiliar."

CALL A.ENTR(LM, LF, rx, ry, rz, kx, ky, kz, Alfa, P1(), P2(), P3(), P4(), P5(), P6())
CALL B.MTRZ(rx, ry, rz, kx, ky, kz, Alfa, RV())
CALL C.DESARROLLO(P1(), P2(), P3(), P4(), P5(), P6(), RV(), L())

END

```

```
SUB A.ENTR (LM, LF, rx, ry, rz, kx, ky, kz, Alfa, P1(), P2(), P3(), P4(), P5(), P6()) STATIC
```

```
'Entradas
```

```
INPUT "El tamaño de cada lado del triángulo móvil"; LM
INPUT "El tamaño de cada lado del triángulo fijo"; LF
INPUT "Dar el valor las coordenadas del centro del triángulo móvil X, Y, Z"; rx, ry, rz
INPUT "Dar la proyección en X del vector auxiliar"; kx
INPUT "Dar la proyección en Y del vector auxiliar"; ky
INPUT "Dar la proyección en Z del vector auxiliar"; kz
INPUT "Dar el ángulo sobre el cual rotar el sistema de referencia"; Alfa
```

```
Pi = 3.14159265359#
```

```
Alfa = Alfa * Pi / 180
```

```
'Posición de los vértices con respecto al centro de cada triángulo
```

```
P1(1) = -LM / (2 * SQR(3))
```

```
P1(2) = -LM / 2
```

```
P1(3) = 0
```

```
P1(4) = 1
```

```
P2(1) = LM / SQR(3)
```

```
P2(2) = 0
```

```
P2(3) = 0
```

```
P2(4) = 1
```

```
P3(1) = -LM / (2 * SQR(3))
```

```
P3(2) = LM / 2
```

```
P3(3) = 0
```

```
P3(4) = 1
```

```
P4(1) = -LF / (2 * SQR(3))
```

```
P4(2) = LF / 2
```

```
P4(3) = 0
```

```
P5(1) = LF / SQR(3)
```

```
P5(2) = 0
```

```
P5(3) = 0
```

```
P6(1) = -LF / (2 * SQR(3))
```

```
P6(2) = -LF / 2
```

```
P6(3) = 0
```

```
END SUB
```

```
SUB B.MTRZ (rx, ry, rz, kx, ky, kz, Alfa, RV()) STATIC
```

```
CALfa = COS(Alfa)
```

```
SALfa = SIN(Alfa)
```

```
VALfa = 1 - COS(Alfa)
```

```
'Estos valores representan a la matriz de rotación
```

```
RV(1, 1) = kx ^ 2 * VALfa + CALfa
```

```
RV(2, 1) = kx * ky * VALfa + kz * SALfa
```

```
RV(3, 1) = kx * kz * VALfa - ky * SALfa
```

```
RV(1, 2) = kx * ky * VALfa - kz * SALfa
```

```
RV(2, 2) = ky ^ 2 * VALfa + CALfa
```

```
RV(3, 2) = ky * kz * VALfa + kx * SALfa
```

```
RV(1, 3) = kx * kz * VALfa + ky * SALfa
```

```
RV(2, 3) = ky * kz * VALfa - kx * SALfa
```

```
RV(3, 3) = kz ^ 2 * VALfa + CALfa
```

```
'Estos valores representan al vector de posición
```

```
RV(1, 4) = rx
```

```
RV(2, 4) = ry
```

```
RV(3, 4) = rz
```

```
'Los valores de la transformación de perspectiva
```

```
RV(4, 1) = 0
```

```
RV(4, 2) = 0
```

```
RV(4, 3) = 0
```

```
RV(4, 4) = 1
```

```

'Impresión de la matriz con el vector auxiliar
PRINT
PRINT " La matriz de transformación es:"
PRINT
FOR i = 1 TO 3
  FOR j = 1 TO 4
    PRINT " "; RV(i, j); " ";
  NEXT j
  PRINT
NEXT i
PRINT " "; RV(4, 1), " "; RV(4, 2), " "; RV(4, 3), " "; RV(4, 4)
END SUB

```

**SUB C.DESARROLLO (P1(), P2(), P3(), P4(), P5(), P6(), RV(), L()) STATIC**

'Obtención de la longitud de los actuadores

```

L(1, 1) = RV(1, 1) * P2(1) + RV(1, 2) * P2(2) + RV(1, 3) * P2(3) + RV(1, 4) * P2(4) + P6(1)
L(1, 2) = RV(2, 1) * P2(1) + RV(2, 2) * P2(2) + RV(2, 3) * P2(3) + RV(2, 4) * P2(4) + P6(2)
L(1, 3) = RV(3, 1) * P2(1) + RV(3, 2) * P2(2) + RV(3, 3) * P2(3) + RV(3, 4) * P2(4) + P6(3)

L(2, 1) = RV(1, 1) * P2(1) + RV(1, 2) * P2(2) + RV(1, 3) * P2(3) + RV(1, 4) * P2(4) + P4(1)
L(2, 2) = RV(2, 1) * P2(1) + RV(2, 2) * P2(2) + RV(2, 3) * P2(3) + RV(2, 4) * P2(4) + P4(2)
L(2, 3) = RV(3, 1) * P2(1) + RV(3, 2) * P2(2) + RV(3, 3) * P2(3) + RV(3, 4) * P2(4) + P4(3)

L(3, 1) = RV(1, 1) * P1(1) + RV(1, 2) * P1(2) + RV(1, 3) * P1(3) + RV(1, 4) * P1(4) + P4(1)
L(3, 2) = RV(2, 1) * P1(1) + RV(2, 2) * P1(2) + RV(2, 3) * P1(3) + RV(2, 4) * P1(4) + P4(2)
L(3, 3) = RV(3, 1) * P1(1) + RV(3, 2) * P1(2) + RV(3, 3) * P1(3) + RV(3, 4) * P1(4) + P4(3)

L(4, 1) = RV(1, 1) * P1(1) + RV(1, 2) * P1(2) + RV(1, 3) * P1(3) + RV(1, 4) * P1(4) + P5(1)
L(4, 2) = RV(2, 1) * P1(1) + RV(2, 2) * P1(2) + RV(2, 3) * P1(3) + RV(2, 4) * P1(4) + P5(2)
L(4, 3) = RV(3, 1) * P1(1) + RV(3, 2) * P1(2) + RV(3, 3) * P1(3) + RV(3, 4) * P1(4) + P5(3)

L(5, 1) = RV(1, 1) * P3(1) + RV(1, 2) * P3(2) + RV(1, 3) * P3(3) + RV(1, 4) * P3(4) + P5(1)
L(5, 2) = RV(2, 1) * P3(1) + RV(2, 2) * P3(2) + RV(2, 3) * P3(3) + RV(2, 4) * P3(4) + P5(2)
L(5, 3) = RV(3, 1) * P3(1) + RV(3, 2) * P3(2) + RV(3, 3) * P3(3) + RV(3, 4) * P3(4) + P5(3)

L(6, 1) = RV(1, 1) * P3(1) + RV(1, 2) * P3(2) + RV(1, 3) * P3(3) + RV(1, 4) * P3(4) + P6(1)
L(6, 2) = RV(2, 1) * P3(1) + RV(2, 2) * P3(2) + RV(2, 3) * P3(3) + RV(2, 4) * P3(4) + P6(2)
L(6, 3) = RV(3, 1) * P3(1) + RV(3, 2) * P3(2) + RV(3, 3) * P3(3) + RV(3, 4) * P3(4) + P6(3)

```

```

PRINT
PRINT
PRINT

```

LOCATE 12, 1: PRINT " Vértice 1 Vértice 2 Vértice 3"

```

FOR i = 1 TO 3
  LOCATE 13 + i, 2: PRINT L(3, i) - P4(i)
  LOCATE 13 + i, 15: PRINT L(1, i) - P6(i)
  LOCATE 13 + i, 28: PRINT L(5, i) - P5(i)
PRINT

```

```

NEXT i
PRINT

```

```

Ac1 = SQR(L(1, 1) ^ 2 + L(1, 2) ^ 2 + L(1, 3) ^ 2)
Ac2 = SQR(L(2, 1) ^ 2 + L(2, 2) ^ 2 + L(2, 3) ^ 2)
Ac3 = SQR(L(3, 1) ^ 2 + L(3, 2) ^ 2 + L(3, 3) ^ 2)
Ac4 = SQR(L(4, 1) ^ 2 + L(4, 2) ^ 2 + L(4, 3) ^ 2)
Ac5 = SQR(L(5, 1) ^ 2 + L(5, 2) ^ 2 + L(5, 3) ^ 2)
Ac6 = SQR(L(6, 1) ^ 2 + L(6, 2) ^ 2 + L(6, 3) ^ 2)

```

```

PRINT " Longitud del Actuador 1 = "; Ac1
PRINT " Longitud del Actuador 2 = "; Ac2
PRINT " Longitud del Actuador 3 = "; Ac3
PRINT " Longitud del Actuador 4 = "; Ac4
PRINT " Longitud del Actuador 5 = "; Ac5
PRINT " Longitud del Actuador 6 = "; Ac6

```

END SUB

## Dinámica

```

DECLARE SUB A1.Entr.L (L1, L2, L3, L4, L5, L6, L(), LM, LP)
DECLARE SUB A2.ENTR (LM, LP, P4(), P5(), P6(), fil, fi2, fi3)
DECLARE SUB B.TRIG (LP, L(), V1(), V2(), V3(), r1, r2, r3)
DECLARE SUB C.CICLO (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, m, k, LM, r1, r2, r3, fil,
    fi2, fi3, hn1, L12min, L12max, L13min, L13max, L(), LM(), V1(), V2(),
    V3(), P1(), P2(), P3(), P4(), P5(), P6(), Pd1())

DECLARE SUB C1.Vert1 (Theta1, r1, fil, hn1, V1(), P1(), Pd1())
DECLARE SUB C2.IT13 (Theta1, Theta3a, Theta3b, LM, r1, r3, fil, fi3, hn1, L13min, L13max,
    V1(), V3(), P5(), P6(), P1(), P2(), P3(), Pd1())
DECLARE SUB C3.IT12 (Theta1, Theta2a, Theta2b, LM, r1, r2, fil, fi2, hn1, L12min, L12max,
    L13min, L13max, V1(), V2(), P4(), P6(), P1(), P2(), P3(), Pd1())
DECLARE SUB C4.Conv23 (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, LM(), L12min,
    L12max, L13min, L13max, P2(), P3())
DECLARE SUB CS.POSIC (k, L(), P1(), P2(), P3(), P4(), P5(), P6(), P1x, P2x, P3x, P1y, P2y,
    P3y, P1z, P2z, P3z)

DECLARE SUB D.VECT.GRAV (Vgx, Vgy, Vgz)
DECLARE SUB E.Dinámica (Act, L(), P1(), P2(), P3(), P4(), P5(), P6())

CLS

DIM L(10)
DIM V1(1000)
DIM V2(1000)
DIM V3(1000)
DIM P4(1000)
DIM P5(1000)
DIM P6(1000)
DIM P1(1000)
DIM P2(1000)
DIM P3(1000)

DIM Pd1(100)
DIM LM(100, 100)

'Primer ciclo
CALL A1.Entr.L(L1, L2, L3, L4, L5, L6, L(), LM, LP)
CALL A2.ENTR(LM, LP, P4(), P5(), P6(), fil, fi2, fi3)
CALL B.TRIG(LP, L(), V1(), V2(), V3(), r1, r2, r3)
CALL C.CICLO(Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, r1, r2, r3, fil,
    fi2, fi3, hn1, L12min, L12max, L13min, L13max, L(), LM(), V1(), V2(), V3(),
    P1(), P2(), P3(), P4(), P5(), P6(), Pd1())

P1xi = P1(1): P1yi = P1(2): P1zi = P1(3)
P2xi = P2(1): P2yi = P2(2): P2zi = P2(3)
P3xi = P3(1): P3yi = P3(2): P3zi = P3(3)

'Segundo ciclo
CALL E.Dinámica(Act, L(), P1(), P2(), P3(), P4(), P5(), P6())
CALL A2.ENTR(LM, LP, P4(), P5(), P6(), fil, fi2, fi3)
CALL B.TRIG(LP, L(), V1(), V2(), V3(), r1, r2, r3)
CALL C.CICLO(Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, r1, r2, r3, fil,
    fi2, fi3, hn1, L12min, L12max, L13min, L13max, L(), LM(), V1(), V2(), V3(),
    P1(), P2(), P3(), P4(), P5(), P6(), Pd1())
CALL D.VECT.GRAV(Vgx, Vgy, Vgz)

dP1(1) = P1(1) - P1xi: dP1(2) = P1(2) - P1yi: dP1(3) = P1(3) - P1zi
dP2(1) = P2(1) - P2xi: dP2(2) = P2(2) - P2yi: dP2(3) = P2(3) - P2zi
dP3(1) = P3(1) - P3xi: dP3(2) = P3(2) - P3yi: dP3(3) = P3(3) - P3zi

'Incrementos de cada vértice
PRINT
PRINT "      dP1(x,y,z)", "      dP2(x, y, z) ", "      dP3(x, y, z)"
PRINT
PRINT " "; dP1(1), "      "; dP2(1), " "; dP3(1)
PRINT " "; dP1(2), "      "; dP2(2), " "; dP3(2)
PRINT " "; dP1(3), "      "; dP2(3), " "; dP3(3)

```

```

'Incrementos de los actuadores
dL(1) = L(1) - L1
dL(2) = L(2) - L2
dL(3) = L(3) - L3
dL(4) = L(4) - L4
dL(5) = L(5) - L5
dL(6) = L(6) - L6

'Incrementos del centro del triángulo
dPx(Act) = (dP1(1) + dP2(1) + dP3(1)) / 3
dPy(Act) = (dP1(2) + dP2(2) + dP3(2)) / 3
dPz(Act) = (dP1(3) + dP2(3) + dP3(3)) / 3

'Incremento del centro del triángulo en dirección del vector gravedad
dPgX(Act) = dPx(Act) * Vgx
dPgy(Act) = dPy(Act) * Vgy
dPgZ(Act) = dPz(Act) * Vgz
dPg(Act) = dPgX(Act) + dPgy(Act) + dPgZ(Act)

'Impresión
PRINT
PRINT " El incremento en dirección de la gravedad dPg (*; dPgX(Act); dPgy(Act); dPgZ(Act);
*) "
PRINT
INPUT " ¿Cuál es el peso del triángulo móvil"; Peso
PRINT

IF dL(Act) = 0 THEN
    PRINT "No hubo desplazamiento alguno"
END
END IF

'Cálculo de la fuerza
fa(Act) = Peso * dPg(Act) / dL(Act)
PRINT "La fuerza requerida para poder desplazarse el actuador"; Act; "es "; fa(Act)

END

SUB E.Dinámica (Act, L(), P1(), P2(), P3(), P4(), P5(), P6()) STATIC
DO
    INPUT "¿Qué actuador quieres analizar"; Act
    Act = INT(Act)
    LOOP UNTIL Act >= 1 AND Act <= 6

    INPUT "¿Cuál es la nueva posición"; Posición

    SELECT CASE Act
        CASE 1: L(1) = Posición
        CASE 2: L(2) = Posición
        CASE 3: L(3) = Posición
        CASE 4: L(4) = Posición
        CASE 5: L(5) = Posición
        CASE 6: L(6) = Posición
    END SELECT
END SUB

```

## Control (Triángulo de Referencia)

```

DECLARE SUB A.ENTR (LM, LF, L(), P4(), P5(), P6(), fi1, fi2, fi3)
DECLARE SUB B.TRIG (LF, L(), V1(), V2(), V3(), r1, r2, r3)
DECLARE SUB C.CICLO (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, m, k, LM, r1, r2, r3, fi1,
                    fi2, fi3, hn1, L12min, L12max, L13min, L13max, L(), LM(), V1(), V2(),
                    V3(), P1(), P2(), P3(), P4(), P5(), P6(), Pd1())

DECLARE SUB C1.Vert1 (Theta1, r1, fi1, hn1, V1(), P1(), Pd1())
DECLARE SUB C2.IT13 (Theta1, Theta3a, Theta3b, LM, r1, r3, fi1, fi3, hn1, L13min, L13max,
                    V1(), V3(), P5(), P6(), P1(), P2(), P3(), Pd1())
DECLARE SUB C3.IT12 (Theta1, Theta2a, Theta2b, LM, r1, r2, fi1, fi2, hn1, L12min, L12max,
                    L13min, L13max, V1(), V2(), P4(), P6(), P1(), P2(), P3(), Pd1())
DECLARE SUB C4.Conv23 (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, LM(), L12min,
                    L12max, L13min, L13max, P2(), P3())
DECLARE SUB C5.POSIC (k, P1(), P2(), P3(), P4(), P5(), P6(), Plx, P2x, P3x, Ply, P2y, P3y,
                    Plz, P2z, P3z, L())

DECLARE SUB C51.VECT (rx, ry, rz, Plx, P2x, P3x, Ply, P2y, P3y, Plz, P2z, P3z)
DECLARE SUB C52.MATRIZ (rx, ry, rz, Plx, P2x, P3x, Ply, P2y, P3y, Plz, P2z, P3z, P4(), P5(),
                    P6(), R(), RM(), L())

DECLARE SUB D.REFERENCIA (P1(), P2(), P3(), P4(), P5(), P6(), LF, LM)
DECLARE SUB D1.Matriz (Vpx, Vpy, Vpz, Psi, Theta, Phi, RE())
DECLARE SUB D2.Puntos (P1(), P2(), P3(), P4(), P5(), P6(), RE(), LP, LM)

DECLARE SUB E.IMAGEN (P1(), P2(), P3(), Imx(), Imy(), Imz(), PCx(), PCy(), LM, Lmi, L())
DECLARE SUB E1.IMPRES (Imx(), Imy(), Imz(), PCx(), PCy(), Lmi, L())
DECLARE SUB E11.EJES (A, B)
DECLARE SUB E12.PUNTCOND (Imx(), Imy(), Imz(), PCx(), PCy(), Lmi)

DECLARE SUB F1.IMPRTXT (Imx(), Imy(), PCx(), PCy(), L())

DECLARE SUB G.DESPLAZ (P1(), P2(), P3(), P4(), P5(), P6(), R(), RM(), L(), DTheta,
                    SDTheta(), k)

DECLARE SUB H1.DESP.XY (Vdx, Vdy, P1(), P2(), P3(), P4(), P5(), P6(), L(), Imx(), Imy(),
                    Imz(), PCx(), PCy(), LM, Lmi)
DECLARE SUB H2.ROT.Z (angT, P1(), P2(), P3(), P4(), P5(), P6(), L(), LM, Imx(), Imy())
DECLARE SUB H3.ENFOQUE (P1(), P2(), P3(), P4(), P5(), P6(), L(), DEnf, Imx(), Imy(), Imz(),
                    PCx(), PCy(), LM, Lmi)

DECLARE SUB I.TEST (P1(), P2(), P3(), LM)

CLS
SCREEN 12

DIM L(10)
DIM V1(1000): DIM V2(1000): DIM V3(1000)
DIM P4(33): DIM P5(33): DIM P6(33)
DIM P1(33): DIM P2(33): DIM P3(33)
DIM Pd1(1000)
DIM LM(100, 100)
DIM R(4, 4): DIM RM(4, 4): DIM RE(4, 4)

DIM Imx(3): DIM Imy(3): DIM Imz(3)
DIM PCx(3): DIM PCy(3)
DIM Dist(3)
DIM Dst(3)
DIM SDTheta(3)

' Cinemática Directa
CALL A.ENTR(LM, LF, L(), P4(), P5(), P6(), fi1, fi2, fi3)
CALL B.TRIG(LF, L(), V1(), V2(), V3(), r1, r2, r3)
CALL C.CICLO(Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, r1, r2, r3, fi1, fi2,
            fi3, hn1, L12min, L12max, L13min, L13max, L(), LM(), V1(), V2(), V3(),
            P1(), P2(), P3(), P4(), P5(), P6(), Pd1())
CALL D.REFERENCIA(P1(), P2(), P3(), P4(), P5(), P6(), LF, LM)

```

```

'EMPIEZA EL CONTROL
'Condiciones iniciales
Theta1 = 0
Pi = 3.1415926536#

    P1bx = P1(1): P1by = P1(2): P1bz = P1(3)
    P2bx = P2(1): P2by = P2(2): P2bz = P2(3)
    P3bx = P3(1): P3by = P3(2): P3bz = P3(3)
    P1ax = P1(1): P1ay = P1(2): P1az = P1(3)
    P2ax = P2(1): P2ay = P2(2): P2az = P2(3)
    P3ax = P3(1): P3ay = P3(2): P3az = P3(3)

H2 = 4
H3 = 3

CLS
FOR jj = 0 TO 1000
CLS
    CALL E.IMAGEN(P1(), P2(), P3(), Imx(), Imy(), Imz(), PCx(), PCy(), LM, LMi, L())
    CALL F1.IMPR.TXT(Imx(), Imy(), PCx(), PCy(), L())
    CALL I.TEST(P1(), P2(), P3(), LM)

'Media pesada
    A = .5
    B = 1 - A
    P1(1) = A * P1ax + B * P1bx: P1(2) = A * P1ay + B * P1by: P1(3) = A * P1az + B * P1bz
    P2(1) = A * P2ax + B * P2bx: P2(2) = A * P2ay + B * P2by: P2(3) = A * P2az + B * P2bz
    P3(1) = A * P3ax + B * P3bx: P3(2) = A * P3ay + B * P3by: P3(3) = A * P3az + B * P3bz

    LOCATE 10, 65: COLOR 15: PRINT "Algoritmo"
    LOCATE 1, 37: PRINT "Ciclo"; jj: COLOR 7

'Desplazamiento XY
    CALL H1.DESP.XY(vdx, vdy, P1(), P2(), P3(), P4(), P5(), P6(), L(), Imx(), Imy(),
        Imz(), PCx(), PCy(), LM, LMi)
    LOCATE 11, 65: PRINT "Deplaz XY"

'Rotación sobre el eje Z
    IF jj = H2 THEN
        H2 = jj + 3
        CALL H2.ROT.Z(angT, P1(), P2(), P3(), P4(), P5(), P6(), L(), LM, Imx(), Imy())
        LOCATE 12, 65: PRINT "Rotación Z"
    END IF

'Enfoque, desplazamiento sobre el eje Z
    IF jj = H3 AND jj < 25 THEN
        H3 = jj + 3
        CALL H3.ENFOQUE(P1(), P2(), P3(), P4(), P5(), P6(), L(), DEnf, Imx(), Imy(), Imz(),
            PCx(), PCy(), LM, LMi)
        LOCATE 13, 65: PRINT "Enfoque"
    END IF

'Valores iniciales de los puntos antes de entrar al algoritmo primario
    P1bx = P1(1): P1by = P1(2): P1bz = P1(3)
    P2bx = P2(1): P2by = P2(2): P2bz = P2(3)
    P3bx = P3(1): P3by = P3(2): P3bz = P3(3)

'Desplazamiento de los vértices

FOR k = 1 TO 3

'Condiciones iniciales
    IF jj = 1 THEN
        DTheta = -1
    ELSE
        DTheta = -ABS(SDTheta(k)) * 100
        IF DTheta < -1 THEN
            DTheta = -1
        END IF
    END IF
    SDTheta(k) = 0

```

```

'Inicio de la iteración para determinar la mínima distancia
FOR j = 0 TO 1000
  'Estado inicial
  Dst(k) = Dist(k)
  'Desplazamiento de los vértices y su proyección
  CALL G.DESPLAZ(P1(), P2(), P3(), P4(), P5(), P6(), R(), RM(), L(), DTheta,
    SDTheta(), k)
    IF k = 1 THEN P1ax = P1(1): P1ay = P1(2): P1az = P1(3)
    IF k = 2 THEN P2ax = P2(1): P2ay = P2(2): P2az = P2(3)
    IF k = 3 THEN P3ax = P3(1): P3ay = P3(2): P3az = P3(3)
  CALL E.IMAGEN(P1(), P2(), P3(), Imx(), Imy(), Imz(), PCx(), PCy(), LM, LMi, L())
  CALL I.TEST(P1(), P2(), P3(), LM)
  'Función de comparación
  Dist(k) = SQR((Imx(k) - PCx(k)) ^ 2 + (Imy(k) - PCy(k)) ^ 2)
  Dif1 = Dif
  Dif = Dist(k) - Dst(k)
  'Variación del ángulo DTheta
  IF jj < 20 THEN
    paso = .1
  ELSE
    paso = .3
  END IF
  IF Dif > 0 THEN
    IF j > 1 THEN
      DTheta = -paso * DTheta
    ELSE
      DTheta = -DTheta
    END IF
  END IF
  'Condición de salida del ciclo
  Diferencia = Dif1 - Dif
  IF Diferencia = 0 THEN EXIT FOR
NEXT j
  'Toma los valores iniciales
  P1(1) = P1bx: P1(2) = P1by: P1(3) = P1bz
  P2(1) = P2bx: P2(2) = P2by: P2(3) = P2bz
  P3(1) = P3bx: P3(2) = P3by: P3(3) = P3bz
NEXT k
'Condición de salida
Dist1 = SQR((Imx(1) - PCx(1)) ^ 2 + (Imy(1) - PCy(1)) ^ 2)
Dist2 = SQR((Imx(2) - PCx(2)) ^ 2 + (Imy(2) - PCy(2)) ^ 2)
Dist3 = SQR((Imx(3) - PCx(3)) ^ 2 + (Imy(3) - PCy(3)) ^ 2)
DistFinal = Dist1 ^ 2 + Dist2 ^ 2 + Dist3 ^ 2
IF DistFinal < .000000001# THEN EXIT FOR
NEXT jj
CALL F1.IMPR.TXT(Imx(), Imy(), PCx(), PCy(), L())
LOCATE 20, 1: PRINT "Terminó"
END

```

## Control (Circunferencia de Referencia)

```

DECLARE SUB A.ENTR (LM, LF, L(), P4(), P5(), P6(), fil, fi2, fi3)
DECLARE SUB B.TRIG (LF, L(), V1(), V2(), V3(), r1, r2, r3)
DECLARE SUB C.CICLO (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, m, k, LM, r1, r2, r3, fil,
fi2, fi3, hnl, L12min, L12max, L13min, L13max, L(), LM(), V1(), V2(),
V3(), P1(), P2(), P3(), P4(), P5(), P6(), Pdl())
DECLARE SUB C1.Vert1 (Theta1, r1, fil, hnl, V1(), P1(), Pdl())
DECLARE SUB C2.IT13 (Theta1, Theta3a, Theta3b, LM, r1, r3, fil, fi3, hnl, L13min, L13max,
V1(), V3(), P5(), P6(), P1(), P2(), P3(), Pdl())
DECLARE SUB C3.IT12 (Theta1, Theta2a, Theta2b, LM, r1, r2, fil, fi2, hnl, L12min, L12max,
L13min, L13max, V1(), V2(), P4(), P6(), P1(), P2(), P3(), Pdl())
DECLARE SUB C4.Conv23 (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, LM(), L12min,
L12max, L13min, L13max, P2(), P3())
DECLARE SUB C5.POSIC (k, P1(), P2(), P3(), P4(), P5(), P6(), P1x, P2x, P3x, P1y, P2y, P3y,
P1z, P2z, P3z, L())
DECLARE SUB C51.VECT (rx, ry, rz, P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z)
DECLARE SUB C52.MATRIZ (rx, ry, rz, P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z, P4(), P5(),
P6(), R(), RM(), L())

DECLARE SUB D.REFERENCIA (P1(), P2(), P3(), P4(), P5(), P6(), LF, LM)

DECLARE SUB D1.Matriz (Vpx, Vpy, Vpz, Psi, Theta, Phi, RE())
DECLARE SUB D2.Puntos (P1(), P2(), P3(), P4(), P5(), P6(), RE(), LF, LM)
DECLARE SUB E.IMAGEN (P1(), P2(), P3(), Imx(), Imy(), Imz(), PCx(), PCy(), LM, LMI, L())
DECLARE SUB E1.IMPRES (Imx(), Imy(), Imz(), PCx(), PCy(), LMI, L())
DECLARE SUB E11.EJES (A, B)
DECLARE SUB E12.PUNTCOND (Imx(), Imy(), Imz(), PCx(), PCy(), LMI)
DECLARE SUB F1.IMPR.TXT (Imx(), Imy(), PCx(), PCy(), L(), LMI)

DECLARE SUB G.DESPLAZ (P1(), P2(), P3(), P4(), P5(), P6(), R(), RM(), L(), DTheta,
SDTheta(), k)
DECLARE SUB H1.DESP.XY (Vdx, Vdy, P1(), P2(), P3(), P4(), P5(), P6(), L(), Imx(), Imy(),
Imz(), PCx(), PCy(), LM, LMI)
DECLARE SUB H2.ROT.Z (angT, P1(), P2(), P3(), P4(), P5(), P6(), L(), LM, Imx(), Imy())
DECLARE SUB H3.ENFOQUE (P1(), P2(), P3(), P4(), P5(), P6(), L(), DEnf, Imx(), Imy(), Imz(),
PCx(), PCy(), LM, LMI)
DECLARE SUB H4.CIRC (VDCirx(), VDCiry(), k, P1(), P2(), P3(), P4(), P5(), P6(), L(), Imx(),
Imy(), Imz(), PCx(), PCy(), LM, LMI)
DECLARE SUB I.TEST (P1(), P2(), P3(), LM)

CLS
SCREEN 12
DIM L(10)
DIM V1(1000): DIM V2(1000): DIM V3(1000)
DIM P4(33): DIM P5(33): DIM P6(33)
DIM P1(33): DIM P2(33): DIM P3(33)
DIM Pdl(1000)
DIM LM(100, 100)
DIM R(4, 4): DIM RM(4, 4): DIM RE(4, 4)

DIM Imx(3): DIM Imy(3): DIM Imz(3)
DIM PCx(3): DIM PCy(3)
DIM Dist(3)
DIM Dst(3)
DIM SDTheta(3)
DIM VDCirx(3): DIM VDCiry(3)

'Cinematica Directa
CALL A.ENTR(LM, LF, L(), P4(), P5(), P6(), fil, fi2, fi3)
CALL B.TRIG(LF, L(), V1(), V2(), V3(), r1, r2, r3)
CALL C.CICLO(Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, r1, r2, r3, fil, fi2,
fi3, hnl, L12min, L12max, L13min, L13max, L(), LM(), V1(), V2(), V3(),
P1(), P2(), P3(), P4(), P5(), P6(), Pdl())

CALL D.REFERENCIA(P1(), P2(), P3(), P4(), P5(), P6(), LF, LM)

```

```

'EMPIEZA EL CONTROL
'Condiciones iniciales
Thetal = 0
Pi = 3.1415926536#
    P1bx = P1(1): P1by = P1(2): P1bz = P1(3)
    P2bx = P2(1): P2by = P2(2): P2bz = P2(3)
    P3bx = P3(1): P3by = P3(2): P3bz = P3(3)
    P1ax = P1(1): P1ay = P1(2): P1az = P1(3)
    P2ax = P2(1): P2ay = P2(2): P2az = P2(3)
    P3ax = P3(1): P3ay = P3(2): P3az = P3(3)
CLS
FOR jj = 0 TO 1000
CLS
    CALL E.IMAGEN(P1(), P2(), P3(), Imx(), Imy(), Imz(), PCx(), PCy(), LM, LMi, L())
    CALL F1.INPR.TXT(Imx(), Imy(), PCx(), PCy(), L(), LMi)
    CALL I.TEST(P1(), P2(), P3(), LM)

    LOCATE 10, 65: COLOR 15: PRINT "Algoritmo"
    LOCATE 1, 37: PRINT "Ciclo"; jj: COLOR 7

'Desplazamiento XY
    CALL H1.DESP.XY(Vdx, Vdy, P1(), P2(), P3(), P4(), P5(), P6(), L(), Imx(), Imy(),
        Imz(), PCx(), PCy(), LM, LMi)
    LOCATE 11, 65: PRINT "Desplaz XY"

'Rotación sobre el eje Z
    angT = 1

    FOR ii = 1 TO 200
        CALL H2.ROT.Z(angT, P1(), P2(), P3(), P4(), P5(), P6(), L(), LM, Imx(), Imy())

        Lnon = L(1) + L(3) + L(5)
        Lpar = L(2) + L(4) + L(6)
        DFinal = DistFinal
        DistFinal = ABS(Lnon - Lpar)

        IF DistFinal > DFinal THEN
            IF ii > 1 THEN
                angT = -.3 * angT
            ELSE
                angT = -angT
            END IF
        END IF

        IF DistFinal = 0 THEN EXIT FOR
    NEXT ii
    LOCATE 12, 65: PRINT "Rotación Z"

'Enfoque, desplazamiento sobre el eje Z
    CALL H3.ENFOQUE(P1(), P2(), P3(), P4(), P5(), P6(), L(), DEnf, Imx(), Imy(),
        Imz(), PCx(), PCy(), LM, LMi)
    LOCATE 13, 65: PRINT "Enfoque"

'Valores iniciales de los puntos antes de entrar al algoritmo primario
    P1bx = P1(1): P1by = P1(2): P1bz = P1(3)
    P2bx = P2(1): P2by = P2(2): P2bz = P2(3)
    P3bx = P3(1): P3by = P3(2): P3bz = P3(3)

'Desplazamiento de los vértices
FOR k = 1 TO 3

'Condiciones iniciales
    IF jj = 1 THEN
        DTheta = -.1
    ELSE
        DTheta = -ABS(SDTheta(k)) * 100
        IF DTheta < -.1 THEN
            DTheta = -.1
        END IF
    END IF
    SDTheta(k) = 0

```

'Inicio de la iteración para determinar la mínima distancia

FOR j = 0 TO 1000

'Estado inicial

  Dst(k) = Dist(k)

'Desplazamiento de los vértices y su proyección

  CALL G.DSSPLAZ(P1(), P2(), P3(), P4(), P5(), P6(), R(), RM(), L(), DTheta,  
    SDTheta(), k)

  IF k = 1 THEN P1ax = P1(1): P1ay = P1(2): P1az = P1(3)

  IF k = 2 THEN P2ax = P2(1): P2ay = P2(2): P2az = P2(3)

  IF k = 3 THEN P3ax = P3(1): P3ay = P3(2): P3az = P3(3)

  CALL E.IMAGEN(P1(), P2(), P3(), Imx(), Imy(), Imz(), PCx(), PCy(), LM, LMi, L())

  CALL I.TEST(P1(), P2(), P3(), LM)

'Función de comparación

  Dist(k) = SQR((Imx(k) - PCx(k)) ^ 2 + (Imy(k) - PCy(k)) ^ 2)

  Dif1 = Dif

  Dif = Dist(k) - Dst(k)

'Equidistancia

  L1 = SQR((Imx(2) - Imx(3)) ^ 2 + (Imy(2) - Imy(3)) ^ 2)

  L2 = SQR((Imx(1) - Imx(3)) ^ 2 + (Imy(1) - Imy(3)) ^ 2)

  L3 = SQR((Imx(2) - Imx(1)) ^ 2 + (Imy(2) - Imy(1)) ^ 2)

  IF k = 1 THEN

    Dif1 = Dif23

    Dif23 = ABS(L2 - L3)

    Dif = Dif1 - Dif23

  END IF

  IF k = 2 THEN

    Dif2 = Dif13

    Dif13 = ABS(L1 - L3)

    Dif = Dif2 - Dif13

  END IF

  IF k = 3 THEN

    Dif3 = Dif12

    Dif12 = ABS(L1 - L2)

    Dif = Dif3 - Dif12

  END IF

'Variación del ángulo DTheta

  IF jj < 20 THEN

    paso = .1

  ELSE

    paso = .3

  END IF

  IF Dif < 0 THEN

    IF j > 1 THEN

      DTheta = -paso \* DTheta

    ELSE

      DTheta = -DTheta

    END IF

  END IF

'Condición de salida del ciclo

  IF Dif = 0 THEN EXIT FOR

NEXT j

'Toma los valores iniciales

  P1(1) = P1bx: P1(2) = P1by: P1(3) = P1bz

  P2(1) = P2bx: P2(2) = P2by: P2(3) = P2bz

  P3(1) = P3bx: P3(2) = P3by: P3(3) = P3bz

NEXT k

```

'Media pesada

  A = .3
  B = 1 - A

  P1(1) = A * P1ax + B * P1bx: P1(2) = A * P1ay + B * P1by: P1(3) = A * P1az + B * P1bz
  P2(1) = A * P2ax + B * P2bx: P2(2) = A * P2ay + B * P2by: P2(3) = A * P2az + B * P2bz
  P3(1) = A * P3ax + B * P3bx: P3(2) = A * P3ay + B * P3by: P3(3) = A * P3az + B * P3bz

'Acercamiento
  FOR k = 1 TO 3
    CALL H4.CIRC(VDCirx(), VDCiry(), k, P1(), P2(), P3(), P4(), P5(), P6(), L(), Imx(),
               Imy(), Imz(), PCx(), PCy(), LM, LMi)
    LOCATE 14, 65: PRINT "Acercamiento"
  NEXT k

'Condición de salida

IF jj > 1 THEN
  L1 = SQR((Imx(2) - Imx(3)) ^ 2 + (Imy(2) - Imy(3)) ^ 2)
  L2 = SQR((Imx(1) - Imx(3)) ^ 2 + (Imy(1) - Imy(3)) ^ 2)
  L3 = SQR((Imx(2) - Imx(1)) ^ 2 + (Imy(2) - Imy(1)) ^ 2)

  DSalida = (192 - L1) ^ 2 + (192 - L2) ^ 2 + (192 - L3) ^ 2
  IF DSalida < .002 THEN EXIT FOR
END IF

NEXT jj

CALL F1.IMPR.TXT(Imx(), Imy(), PCx(), PCy(), L(), LMi)
LOCATE 28, 1: PRINT "Terminó"

END

```

## Control (Triángulo de Referencia)

```

DECLARE SUB A.ENTR (LM, LF, L(), P4(), P5(), P6(), fil, fi2, fi3)
DECLARE SUB B.TRIG (LF, L(), V1(), V2(), V3(), r1, r2, r3)

DECLARE SUB C.CICLO (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, m, k, LM, r1, r2, r3, fil,
                   fi2, fi3, hnl, L12min, L12max, L13min, L13max, L(), LM(), V1(), V2(),
                   V3(), P1(), P2(), P3(), P4(), P5(), P6(), Pd1())
DECLARE SUB C1.Vert1 (Theta1, r1, fil, hnl, V1(), P1(), Pd1())
DECLARE SUB C2.IT13 (Theta1, Theta3a, Theta3b, LM, r1, r3, fil, fi3, hnl, L13min, L13max,
                   V1(), V3(), P5(), P6(), P1(), P2(), P3(), Pd1())
DECLARE SUB C3.IT12 (Theta1, Theta2a, Theta2b, LM, r1, r2, fil, fi2, hnl, L12min, L12max,
                   L13min, L13max, V1(), V2(), P4(), P6(), P1(), P2(), P3(), Pd1())
DECLARE SUB C4.Conv23 (Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, LM(), L12min,
                   L12max, L13min, L13max, P2(), P3())
DECLARE SUB C5.POSIC (k, P1(), P2(), P3(), P4(), P5(), P6(), P1x, P2x, P3x, P1y, P2y, P3y,
                   P1z, P2z, P3z, L())
DECLARE SUB C51.VECT (xx, ry, rz, P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z)
DECLARE SUB C52.MATRIZ (rx, ry, rz, P1x, P2x, P3x, P1y, P2y, P3y, P1z, P2z, P3z, P4(), P5(),
                   P6(), R(), RM(), L())

DECLARE SUB D.REFERENCIA (P1(), P2(), P3(), P4(), P5(), P6(), LF, LM)
DECLARE SUB D1.Matriz (Vpx, Vpy, Vpz, Ps1, Theta, Phi, RS())
DECLARE SUB D2.Puntos (P1(), P2(), P3(), P4(), P5(), P6(), RE(), LF, LM)

DECLARE SUB E.IMAGEN (P1(), P2(), P3(), Imx(), Imy(), Imz(), PCx(), PCy(), LM, LMi, L())
DECLARE SUB E1.IMPRES (Imx(), Imy(), Imz(), PCx(), PCy(), LMi, L())
DECLARE SUB E11.EJES (A, B)
DECLARE SUB E12.PUNTCOND (Imx(), Imy(), Imz(), PCx(), PCy(), LMi)

DECLARE SUB F1.IMPR.TXT (Imx(), Imy(), PCx(), PCy(), L())

```

```

DECLARE SUB G.DESPLAZ (P1(), P2(), P3(), P4(), P5(), P6(), R(), RM(), L(), DTheta,
SDTheta(), k)

DECLARE SUB H1.DESP.XY (Vdx, Vdy, P1(), P2(), P3(), P4(), P5(), P6(), L(), Imx(), Imy(),
Imz(), PCx(), PCy(), LM, LMI)
DECLARE SUB H2.ROT.Z (angT, P1(), P2(), P3(), P4(), P5(), P6(), L(), LM, Imx(), Imy())
DECLARE SUB H3.ENFOQUE (P1(), P2(), P3(), P4(), P5(), P6(), L(), DEnf, Imx(), Imy(), Imz(),
PCx(), PCy(), LM, LMI)
DECLARE SUB I.TEST (P1(), P2(), P3(), LM, Imxtest, Imytest)

```

CLS

SCREEN 12

```

DIM L(10)
DIM V1(1000): DIM V2(1000): DIM V3(1000)
DIM P4(33): DIM P5(33): DIM P6(33)
DIM P1(33): DIM P2(33): DIM P3(33)
DIM Pd1(1000)
DIM LM(100, 100)
DIM R(4, 4): DIM RM(4, 4): DIM RE(4, 4)

```

```

DIM Imx(3): DIM Imy(3): DIM Imz(3)
DIM PCx(3): DIM PCy(3)
DIM Dist(3)
DIM Dst(3)
DIM SDTheta(3)

```

'Cinematíca Directa

```

CALL A.ENTR(LM, LP, L(), P4(), P5(), P6(), fi1, fi2, fi3)
CALL B.TRIG(LF, L(), V1(), V2(), V3(), r1, r2, r3)
CALL C.CICLO(Theta1, Theta2a, Theta2b, Theta3a, Theta3b, LM, m, k, r1, r2, r3, fi1, fi2,
fi3, hnl, L12min, L12max, L13min, L13max, L(), LM(), V1(), V2(), V3(),
P1(), P2(), P3(), P4(), P5(), P6(), Pd1())

```

```
CALL D.REFERENCIA(P1(), P2(), P3(), P4(), P5(), P6(), LP, LM)
```

'EMPIEZA EL CONTROL

'Condiciones iniciales

```

Theta1 = 0
Pi = 3.1415926536#

```

```

P1bx = P1(1): P1by = P1(2): P1bz = P1(3)
P2bx = P2(1): P2by = P2(2): P2bz = P2(3)
P3bx = P3(1): P3by = P3(2): P3bz = P3(3)

```

```

P1ax = P1(1): P1ay = P1(2): P1az = P1(3)
P2ax = P2(1): P2ay = P2(2): P2az = P2(3)
P3ax = P3(1): P3ay = P3(2): P3az = P3(3)

```

H2 = 4

H3 = 3

CLS

FOR jj = 0 TO 1000

CLS

```

CALL E.IMAGEN(P1(), P2(), P3(), Imx(), Imy(), Imz(), PCx(), PCy(), LM, LMI, L())
CALL F1.IMPR.TXT(Imx(), Imy(), PCx(), PCy(), L())
CALL I.TEST(P1(), P2(), P3(), LM, Imxtest, Imytest)

```

'Media pasada

```

A = .5
B = 1 - A

```

```

P1(1) = A * P1ax + B * P1bx: P1(2) = A * P1ay + B * P1by: P1(3) = A * P1az + B * P1bz
P2(1) = A * P2ax + B * P2bx: P2(2) = A * P2ay + B * P2by: P2(3) = A * P2az + B * P2bz
P3(1) = A * P3ax + B * P3bx: P3(2) = A * P3ay + B * P3by: P3(3) = A * P3az + B * P3bz

```

```

LOCATE 10, 65: COLOR 15: PRINT "Algoritmo"
LOCATE 1, 37: PRINT "Ciclo"; jj: COLOR 7

```

```

'Desplazamiento XY
CALL H1.DESP.XY(Vdx, Vdy, P1(), P2(), P3(), P4(), P5(), P6(), L(), Imx(), Imy(),
               Imz(), PCx(), PCy(), LM, LMi)
LOCATE 11, 65: PRINT "Deplaz XY"

'Rotación sobre el eje Z
IF jj = H2 THEN
  H2 = jj + 3
CALL H2.ROT.Z(angT, P1(), P2(), P3(), P4(), P5(), P6(), L(), LM, Imx(), Imy())
LOCATE 12, 65: PRINT "Rotación Z"
END IF

'Enfoque, desplazamiento sobre el eje Z
IF jj = H3 AND jj < 25 THEN
  H3 = jj + 3
CALL H3.ENFOQUE(P1(), P2(), P3(), P4(), P5(), P6(), L(), DEnf, Imx(), Imy(), Imz(),
               PCx(), PCy(), LM, LMi)
LOCATE 13, 65: PRINT "Enfoque"
END IF

'Valores iniciales de los puntos antes de entrar al algoritmo primario
P1bx = P1(1): P1by = P1(2): P1bz = P1(3)
P2bx = P2(1): P2by = P2(2): P2bz = P2(3)
P3bx = P3(1): P3by = P3(2): P3bz = P3(3)

'Desplazamiento de los vértices
FOR k = 1 TO 3

'Condiciones iniciales
IF jj = 1 THEN
  DTheta = -1
ELSE
  DTheta = -ABS(SDTheta(k)) * 100
  IF DTheta < -1 THEN
    DTheta = -1
  END IF
END IF

SDTheta(k) = 0

'Inicio de la iteración para determinar la mínima distancia
FOR j = 0 TO 1000

'Estado inicial
Dst = Dist

'Desplazamiento de los vértices y su proyección
CALL G.DESPLAZ(P1(), P2(), P3(), P4(), P5(), P6(), R(), RM(), L(), DTheta,
              SDTheta(), k)

IF k = 1 THEN P1ax = P1(1): P1ay = P1(2): P1az = P1(3)
IF k = 2 THEN P2ax = P2(1): P2ay = P2(2): P2az = P2(3)
IF k = 3 THEN P3ax = P3(1): P3ay = P3(2): P3az = P3(3)

CALL E.IMAGEN(P1(), P2(), P3(), Imx(), Imy(), Imz(), PCx(), PCy(), LM, LMi, L())
CALL I.TEST(P1(), P2(), P3(), LM, Imxtest, Imytest)

'Función de comparación
PCx = (PCx(1) + PCx(2) + PCx(3)) / 3
PCy = (PCy(1) + PCy(2) + PCy(3)) / 3

Dist = SQR((Imxtest - PCx) ^ 2 + (Imytest - PCy) ^ 2)
Dif1 = Dif
Dif = Dist - Dst

```

```

'Variación del ángulo DTheta
  IF jj < 20 THEN
    paso = .1
  ELSE
    paso = .3
  END IF
  IF Dif > 0 THEN
    IF j > 1 THEN
      DTheta = -paso * DTheta
    ELSE
      DTheta = -DTheta
    END IF
  END IF

'Condición de salida del ciclo
  Diferencia = Dif1 - Dif
  IF Diferencia = 0 THEN EXIT FOR

NEXT j

'Toma los valores iniciales
  P1(1) = P1bx: P1(2) = P1by: P1(3) = P1bz
  P2(1) = P2bx: P2(2) = P2by: P2(3) = P2bz
  P3(1) = P3bx: P3(2) = P3by: P3(3) = P3bz

NEXT k

'Condición de salida
  Dist1 = SQR((Imx(1) - PCx(1)) ^ 2 + (Imy(1) - PCy(1)) ^ 2)
  Dist2 = SQR((Imx(2) - PCx(2)) ^ 2 + (Imy(2) - PCy(2)) ^ 2)
  Dist3 = SQR((Imx(3) - PCx(3)) ^ 2 + (Imy(3) - PCy(3)) ^ 2)

  DistFinal = Dist1 ^ 2 + Dist2 ^ 2 + Dist3 ^ 2
  IF DistFinal < .0022# THEN EXIT FOR

NEXT jj

CLS
LOCATE 1, 37: PRINT "Ciclo"; jj: COLOR 7
CALL E.IMAGEN(P1(), P2(), P3(), Imx(), Imy(), Imz(), PCx(), PCy(), LM, Lmi, L())
CALL F1.IMPR.TXT(Imx(), Imy(), PCx(), PCy(), L())
CALL I.TEST(P1(), P2(), P3(), LM, Imxtest, Imytest)
LOCATE 28, 1: PRINT "Terminó"

END

```

## REFERENCIAS

- [ 1 ] Barrientos A. et al, "Fundamentos de Robótica", McGraw-Hill, Madrid, España, 1997, pg17.
- [ 2 ] "The IWF-Hexaglide", ETH-Zürich, Suiza,  
<http://www.iwf.bepr.ethz.ch/projekte/hexaglide>
- [ 3 ] Stewart D., "A platform with six degrees of freedom", Proc. Inst. Mech. Eng., London, Volume 180, 1965, pp 371-386.
- [ 4 ] Fu K.S. et al, "Robótica: Control, Detección, Visión e Inteligencia", McGraw-Hill, Madrid, España, 1988, pp13-15.
- [ 5 ] Fu K.S. et al, "Robótica: Control, Detección, Visión e Inteligencia", McGraw-Hill, Madrid, España, 1988, pp 23-26.
- [ 6 ] Craig J.J., "Introduction to Robotics, Mechanics and Control", Adison Wesley, 2<sup>da</sup> edición, pp 51-53.
- [ 7 ] Begunov B.N., Zakaznov N.P., "Teoría de Sistemas Ópticos", MIR, Moscú, 1976, pp 338-340
- [ 8 ] Stumpff K., "Astronomía", Feltrinelli, 2<sup>da</sup> edición, Milano, Italia, 1964, pg. 369
- [ 9 ] Storm, Dunlop, Guías Prácticas OMEGA, "Astronomía: Guía del Firmamento para Aficionados", OMEGA, Barcelona, España, 1987, pp 61-79.
- [ 10 ] Newton J., Teece Ph., "Astronomía Amateur", OMEGA, Barcelona, España, 1991, pp 24-29.
- [ 11 ] McCloy D. Harris D. M. J., "Robótica: Una Introducción", Limusa, México, pg. 28.
- [ 12 ] Ogata K., "Ingeniería de Control Moderna", Prentice-Hall Hispanoamericana, 3<sup>ra</sup> edición, México, 1997, pp 2,70.
- [ 13 ] Keller A.M., "Programación en Pascal", McGraw-Hill, México, 1983, pp 2-3.

## **BIBLIOGRAFÍA DE APOYO**

- 1.- Leithold L., "El Cálculo con Geometría Analítica", HARLA, 6ª ed. México, 1992.
- 2.- Solís R. et al, "Geometría Analítica", FI-UNAM Limusa, México, 1984.
- 3.- Joyanes Aguilar L., "Programación en Quick Basic 4.0 / 4.5", McGraw-Hill, México, 1992.
- 4.- Gran Enciclopedia Larrousse, Ed. Planeta, Barcelona, España, 3ª edición, 1980

## **PUBLICACIONES**

- Merlet J.P., 1998 "Determination of 6D Workspaces of a Gough-type 6-DOF Parallel Manipulator", in RoManSy, Paris, (6-9 Julio).
- Merlet J.P., 1998 "Determination of the Presence of Singularities in 6D Workspaces of a Gough Parallel Manipulator", in ARK, pp 39-48.
- Gosselin, C., 1996, "Determination of the Workspace of 6-DOF Parallel Manipulators", *Journal of Applied Mechanical Design*, Vol. 112, pp 331-336.
- Griffis, M. W. and Duffy, J., 1989, "A Forward Displacement Analysis of a Class of Stewart Platforms", *Journal of Robotic Systems*, John Wiley, 6(6), pp 703-720.
- Kumar, V., 1992, "Characterization of Workspaces Parallel Manipulators", *ASME Journal of Mechanical Design*, Vol 114, pp 368-375
- Ji, Z., 1994, "Workspace Analysis of Stewart Platforms via Vertex Space", *Journal of Robotic Systems*, 11(7), pp 631-639.
- Lebert, G., et al, 1995, "Dynamic Analysis and Control of a Stewart Platform Manipulator", *Journal of Robotic Systems*, 10(5), pp 629-655.