



**UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO**

---

DIVISIÓN DE ESTUDIOS DE POSGRADO

FACULTAD DE INGENIERÍA

MODELACIÓN Y SIMULACION CINEMÁTICA DE UN  
ROBOT PUMA CON CUATERNIONES

**T E S I S**

QUE PARA OBTENER EL GRADO DE  
MAESTRO EN INGENIERÍA  
(SISTEMAS MECÁNICOS)

P R E S E N T A :  
**ERNESTO OLVERA ARANZOLO**



DIRECTOR DE TESIS: M. I. EUSEBIO JIMÉNEZ LÓPEZ

COASESOR: DR LUIS REYES ÁVILA

CIUDAD UNIVERSITARIA, D.F.

2007



## **Dedicatoria**

A mi esposa Alicia Pérez Aguilar por su comprensión y paciencia en la terminación de este trabajo.

A mi hijo Ernesto Alejandro Olvera Pérez porque esto sea solo el inicio de un camino a recorrer.

A mi hija Alison Noemí Olvera Pérez por ser parte de lo que alegra mi vida.

A mis padres Sr. Alejo Olvera Ledesma, q e p d, y Sra Juana Aranzolo Lugo por darme lo más preciado que es la vida misma.



## **Agradecimientos**

A México por darme la oportunidad de desarrollarme profesionalmente.

A la UNAM por permitirme ser parte de ella.

A la Universidad Autónoma Chapingo por ser mi alma mater y forjarme como profesionista.

Al Dr. Luis Reyes Ávila por sus conocimientos plasmados en esta trabajo.

Al M. I. Eusebio Jiménez por su apoyo, perseverancia y colaboración en la finalización de éste trabajo.

A mis compañeros y amigos que nunca perdieron la fe en la culminación de mi trabajo de tesis.



## Resumen

En diversos trabajos se ha modelado la cinemática de robot PUMA usando métodos tradicionales como Denavit Hartenberg y matrices homogéneas. Cada método tiene sus ventajas y desventajas. Recientemente, se modeló un robot PUMA usando el álgebra de Cuaterniones. Sin embargo, no fue incluido el análisis y la modelación de trayectorias y no fueron optimizadas las ecuaciones resultantes. En este trabajo de tesis se modela la cinemática de posicionamiento de un robot industrial PUMA 560 utilizando el álgebra de Cuaterniones. Son utilizadas las propiedades fundamentales de dicha álgebra para reducir las expresiones algebraicas obtenidas. Se incluye una trayectoria tipo helicoidal la cual sigue el órgano terminal del robot. La programación y la sistematización computacional del modelo se realizan en la plataforma de cálculo formal Mathematica V5. Los resultados derivados de esta tesis, conducen a la obtención de modelos simplificados lo cual reduce significativamente en número de operaciones computacionales. El modelo de la cinemática inversa confiere un sistema de 12 ecuaciones algebraicas no lineales y 12 incógnitas del tipo polinomial, en tanto el problema directo genera un sistema de 6 ecuaciones y 6 incógnitas. Fue utilizado un método numérico para dar solución al sistema de ecuaciones. Finalmente, los resultados obtenidos, pueden ser aplicados a la modelación dinámica de robots espaciales de 6 o más grados de libertad.

Palabras clave: Cuaterniones, Robótica, Cinemática.

## Abstract

The kinematics of the PUMA robot has been modeled in several works using traditional methods like the Denavit-Hartenberg method and the homogeneous matrix method, each one has its own advantages and disadvantages. Recently the PUMA's model was remade using quaternions, however, neither the analysis nor the trajectories' model were included and the resulting equations haven't been optimized. In this thesis the positioning kinematics for an industrial PUMA 560 robot is described using quaternions' algebra. The fundamental properties of such algebra are exploited in order to reduce the complex algebraic expressions obtained. The trajectory described by the terminal organ of the robot is a helicoid, and it is simulated using Mathematica®. The results derived from this thesis are simplified models, easy to code in any computer language. The inverse kinematics model consists of twelve non-linear algebraic equations and twelve polynomial unknowns, and the direct kinematics model is composed by six equations and six unknowns, these systems were solved using a numerical method. The methodology and results show here could be applied by anybody who wants to model the dynamics of spatial robots with six or more degree of freedom.

Index terms: Quaternions, Robotics, Kinematics

## Tabla de contenido

<b>1 Introduccion</b>	<b>1</b>
1.1 El motivo de la investigacion	1
1.2 Los robots y sus elementos	2
1.3 Resumen de capitulado	10
<b>2. Definicion del problema, restricciones e hipotesis</b>	<b>11</b>
2.1 El problema y las restricciones	11
2.2 Hipotesis	13
2.3 Algunas consideraciones importantes	13
<b>3 El marco teorico</b>	<b>15</b>
3.1 El algebra de Cuaterniones.	15
3.2 Representacion parametrica de las rotaciones	17
3.3 Metodo de modelacion	18
<b>4 Algunas consideraciones de composicion de rotaciones</b>	<b>19</b>
4.1 Las rotaciones entre sistemas de referencia no inercial y móvil	19
4.2 Las rotaciones entre las configuraciones	21
4.3 Generalizacion de la composicion de rotaciones	22
4.4 Demostracion de la composicion de rotaciones	22
<b>5 Modelacion y simulacion de la cinematica del robot PUMA 560</b>	<b>25</b>
5.1 Algunas consideraciones importantes para el desarrollo del modelo	24
5.2 Caracterizacion del multicuerpo en estudio	24
5.3 Modelacion de la configuracion no deformada	26
5.4 Modelacion de la configuracion deformada	30
5.5 Cinematica directa	33
5.5.1 <i>Problema cinematico directo</i>	33
5.5.2 <i>Simulacion de la cinematica directo</i>	33
5.5.3 <i>Explicitacion de los ejes de rotacion</i>	42
5.5.4 <i>Explicitacion de los vectores de posicion</i>	45
5.5.5 <i>Salidas graficas</i>	46
5.6 Cinematica inversa	52
5.6.1 <i>Algunas consideraciones sobre la trayectoria y el desarrollo del problema cinematico inverso</i>	53
5.6.2 <i>Trayectoria de operacion</i>	53
5.6.3 <i>Algunas consideraciones importantes para la formulacion del problema cinematico inverso</i>	55
5.6.4 <i>Formulacion del problema cinematico inverso</i>	58
5.6.5 <i>Simulacion de la cinematica inversa</i>	59
<b>6 Conclusiones</b>	<b>71</b>
<b>7 Referencias</b>	<b>73</b>



# Capítulo 1

## Introducción

A través del presente trabajo de tesis se intentarán cumplir los objetivos siguientes:

- Modelar la cinemática inversa y directa de un robot PUMA 560 usando el álgebra de Cuaterniones y una trayectoria helicoidal.
- Simplificar las ecuaciones usando las propiedades del álgebra de Cuaterniones.
- Programar los modelos obtenidos en Mathematica® V5.

### 1.1 El motivo de la investigación

Lo que motiva a la investigación en este trabajo de tesis se sintetiza de la manera siguiente:

*“Para modelar cadenas cinemáticas planas y espaciales son utilizadas diversas metodologías y herramientas matemáticas. Una de dichas herramientas es el álgebra de Cuaterniones [1,2]. Por ejemplo, en [3], se modeló la cinemática de un robot paralelo plano. Además, en [4] se modeló un robot de dos grados de libertad y en [5] se modeló un robot espacial de 3 grados de libertad usando el álgebra de Cuaterniones. Por otro lado, en [6], se modelaron un robot de 3 gdl y en [7] se modeló un mecanismo de 4 barras utilizando también Cuaterniones.*

*Por otro lado, en un trabajo desarrollado [8] se modela la cinemática de un robot PUMA usando Cuaterniones. En dicho trabajo, no se simplifican las ecuaciones resultantes ni se incluye una trayectoria de operación específica. Lo que motiva a la investigación o indagación en este trabajo de tesis, es modelar la cinemática de posicionamiento del robot PUMA 560 simplificando mediante las propiedades del álgebra de Cuaterniones, los modelos resultantes y, a la vez, incluir una trayectoria tipo helicoidal la cual será seguida por el órgano terminal del robot. Otro objetivo*

*fundamental que persigue este trabajo de tesis es expandir el campo de las aplicaciones de los Cuaterniones derivadas de los trabajos realizados en [1,2] y dejar las experiencias del modelado y la sistematización computacional del robot para futuras aplicaciones.”*

## **1.2 Los robots y sus elementos**

La palabra “robot” proviene de la palabra Checa “ROBOTA” que significa trabajo. Para el “Robot Institute of America”, un robot es un manipulador reprogramable multifuncional diseñado para mover materiales, piezas y dispositivos especializados a través de movimientos variables programados para la realización de una diversidad de tareas [4].

Los robots son importantes porque pueden realizar tareas manipuladoras sencillas tanto ó más eficientemente que el hombre y son capaces de realizarlas en forma repetitiva y de manera incansable, pero lo que verdaderamente los hace útiles e importantes es el hecho de que pueden trabajar en medios ambientes en los que un ser humano puede poner en riesgo su integridad física y aún su vida, tal es el caso por ejemplo: en operaciones de manejo de materiales radioactivos; tareas con sustancias químicas; entre otras. Actualmente los robots son utilizados también en la industria espacial, en simuladores de vuelo, en la industria farmacéutica y en general en la industria de la transformación.

Los robots son dispositivos mecánicos y electrónicos de alta precisión que se dañan fácilmente si no se manejan adecuadamente, además de ser herramientas de trabajo muy caras, por estos motivos es necesario proporcionar a las personas que van a trabajar con ellos un medio confiable que les permita familiarizarse con éstos sin arriesgar la integridad física del robot y menos aún de las personas. De esta manera, es necesario capacitar al personal en simuladores computacionales que tienen la ventaja de ser baratos y seguros, ya que lo peor que le puede suceder a un simulador es dejar de interactuar con el usuario, pero este problema se puede resolver fácilmente reiniciando el simulador.

Un robot posee dos características importantes [4]:

1) Versatilidad, la cual puede definirse como la potencialidad estructural de ejecutar tareas diversas y/o ejecutar una misma tarea de forma diversificada.

2) La adaptabilidad al entorno. Esto es, el robot debe, por sí solo, alcanzar su objetivo (la ejecución de una tarea), a pesar de las perturbaciones imprevistas (pero limitadas) de su entorno a lo largo de la ejecución de una tarea.

Un robot manipulador, en principio, es un dispositivo de propósito general. En la práctica los manipuladores son usualmente diseñados pensando en que cubran al menos una clase amplia de aplicaciones, como podría ser, soldar; manejar materiales; ensamblar; entre otras tareas. Estas aplicaciones en gran parte dictan la elección de varios parámetros de diseño del manipulador, incluyendo su estructura cinemática. Por ejemplo, el ensamblado de tarjetas de circuitos es realizado por un manipulador tipo SCARA [4].

Los manipuladores pueden ser clasificados por muchos criterios, tales como su geometría, estructura cinemática, el tipo de aplicación para la cual fueron diseñados, la manera en que son controlados, etc.

Muchos manipuladores industriales actuales tienen 6 ó menos grados de libertad, son usualmente clasificados cinemáticamente sobre la base del primer brazo ó las primeras 3 juntas, describiendo por separado el órgano terminal ó muñeca. La mayoría de estos manipuladores se clasifican en uno de los siguientes tipos de geometría.

- 1 Articulados.
- 2 Esféricos.
- 3 Tipo SCARA.
- 4 Cilíndricos.
- 5 Cartesianos.

Algunos de los robots industriales más usados son los tipo PUMA y la serie IRB ABB Systems de seis grados de libertad para transporte de piezas y ensamblado de partes, el tipo SCARA de cuatro grados de libertad para operaciones de inspección, robot tipo

MotoMan de cinco grados de libertad para operaciones de soldadura y el tipo AS-RS para trabajos de almacenaje de dos grados de libertad. Existen en la actualidad cadenas cinemáticas especiales llamadas plataformas ó robots paralelos que tiene ventajas sobre los manipuladores convencionales dada su alta precisión, sin embargo, su uso no está bien difundido por lo complicado de sus modelos matemáticos y altas redundancias, por lo que se encuentran en laboratorios de las universidades y su uso esta restringido en la industria manufacturera, sin embargo, en la industria aeronáutica y espacial son usados para construir simuladores complejos. Un simulador de robot es un sistema que imita el funcionamiento ó comportamiento del robot. Tiene el objetivo de enseñar a los futuros operadores las características y restricciones de robot que se va a manejar.

Las figuras 1.1 y 1.2 muestran dos ejemplos de robots industriales.

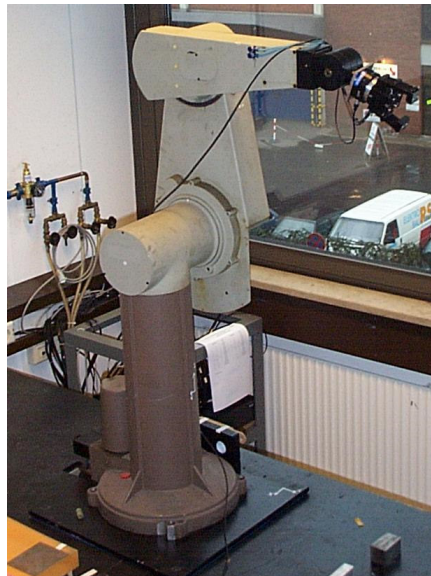


Figura 1.1 Robot industrial tipo PUMA



Figura 1.2 Robot industrial tipo paralelo DELTA

El robot mostrado en la figura 1.1 es de cadena abierta y el mostrado en la figura 1.2 es de cadena paralela o, mejor dicho, es una cadena cinemática híbrida que combina cadenas cerradas y abiertas. De acuerdo con [9], las cadenas cinemáticas se pueden clasificar en: abiertas y cerradas. La figura 1.3 muestra ejemplos de configuración de dichas cadenas:

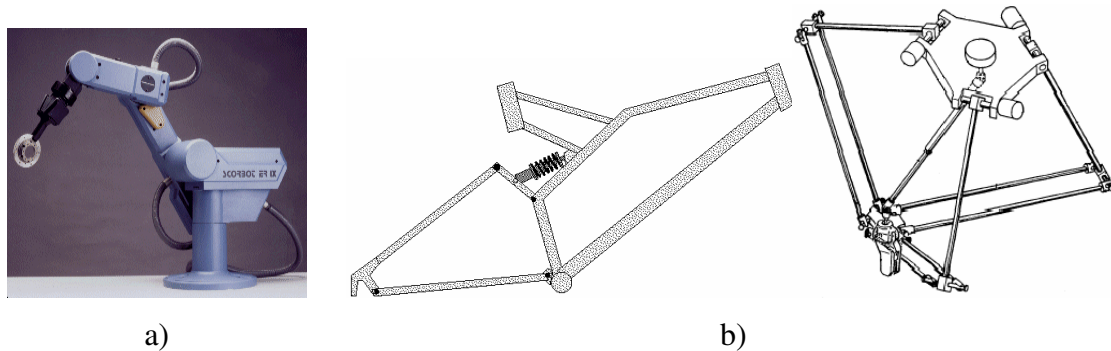


Figura 1.3 Cadenas cinemáticas: a) Abierta y b) Cerrada

Cabe señalar que una cadena cinemática es un sistema formado de eslabones y juntas las cuales prescriben cierto tipo de movimiento. Los robots paralelos, al ser formados por cadenas cerradas, son más fuertes y precisos, sin embargo, el área de trabajo es muy limitada y suelen tener problemas de singularidades [9]. Por otro lado, el número de grados de libertad (GDL) de un sistema o cadena cinemática es el número de parámetros independientes que se requieren para definir unívocamente su posición en el espacio y en cualquier instante o, de manera equivalente, es el número de coordenadas independientes requeridas para definir la posición y orientación de un objeto, por ejemplo, el robot mostrado en la figura 1.1 es de 6 GDL; esto es, 3 de posición y tres de orientación.

Por otro lado, las juntas que conectan los eslabones son de diversos tipos y de diferentes grados de libertad. Por ejemplo, la junta más sencilla es la tipo rotacional, también lo es la junta prismática, pues ambas son de un grado de libertad, en tanto la junta esférica es de tres grados de libertad. La tabla 1.1 muestra las diferentes juntas que existen y los tipos de pares cinemáticos, así como algunos ejemplos de aplicación donde se muestran diversas conexiones [10].

Tipo de junta	Par inferior (I) o par superior (S)	Símbolo	Grados de Libertad	Representación Esquemática	Posible Configuración	Ejemplo Descriptivo
Rotacional	I	R	1 $\theta$			Un pasador fijo que permite solo la rotación
Prismática	I	P	1 x			Una barra recta que solo permite el deslizamiento
Helicoidal	I	H	1 x ó $\theta$			Un tornillo impulsor o una barra helicoidal
Cilíndrica	I	C	2 x ó $\theta$			Un brazo que permite tanto rotación como deslizamiento
Esférica	I	S	3 $\theta, \phi, \gamma$			Una junta esférica que permite rotación en 3 direcciones angulares
Plana	I	P <sub>L</sub>	3 x,y, $\theta$			Una Superficie restringida que permite rotación y movimiento paralelo al plano de la superficie
Junta Universal	I	U	2 $\theta, \phi$			La junta universal Hooke combina 2 pares rotatorios
Engranaje	S	G	2 Rodar y trasladar			Engranajes
Par de Levas	H	*	2 Rodar y trasladar	*		Disco de levas y seguidores

Tabla 1.1 Descripción de juntas y grado de libertad

Por otro lado, el área o volumen de trabajo de un robot, es una configuración espacial en donde el órgano terminal del robot se moverá y realizará su trabajo. Las figuras 1.4 y 1.5 muestran las áreas de trabajo de un robot de cadena abierta y de un robot paralelo plano, respectivamente.



Figura 1.4 Área de trabajo de un robot de cadena abierta

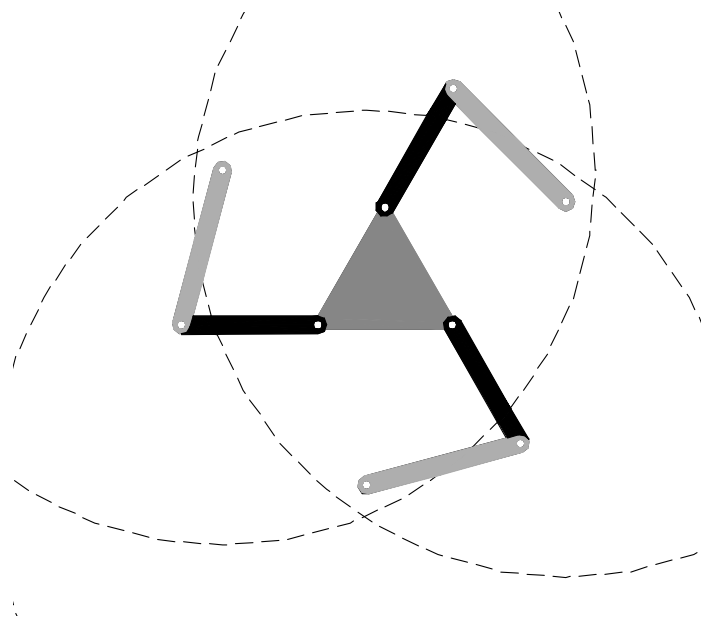


Figura 1.5 Área de trabajo de un robot plano paralelo

Por otro lado, la mayor parte de la maquinaria integrada a un sistema productivo tiene asociada a su estructura un dispositivo o elemento de trabajo, mediante el cual, realizan operaciones de manufactura. Por ejemplo, un robot industrial puede integrar varios cabezales de operación en un órgano terminal y, realizar con ellos, operaciones de soldadura, corte, ensamble y transporte de componentes entre otras tareas.

Las máquinas de control numérico y centros de maquinado poseen un cabezal de múltiples herramientas con las cuales maquinan las piezas de trabajo. Para cada operación de manufactura existe una ruta de trabajo que la herramienta o dispositivo debe seguir para realizar dicha operación. Las rutas de trabajo son, rigurosamente hablando, lugares geométricos descritos en el espacio-tiempo, y pueden ser simples líneas rectas para el caso de maquinados o lugares geométricos complicados como una curva para operaciones de ensamblaje de productos [11].

En cada operación o tarea de trabajo de una máquina, se requiere que la herramienta se traslade a través de un lugar geométrico con diferentes velocidades. Por ejemplo, en un ciclo simple de maquinado asociado con un CNC, se requiere que la herramienta de corte se desplace desde un origen hasta la pieza de trabajo a una cierta velocidad. Posteriormente, la herramienta realiza la operación de corte de material a velocidad constante (velocidad de corte) y, finalmente, una vez terminada la operación, debe regresar al punto de partida a cierta velocidad. La herramienta de trabajo en cada punto o sección del lugar geométrico tiene asociados tres parámetros: desplazamiento, velocidad y aceleración.

La figura 1.6 muestra algunos lugares geométricos relacionados con una trayectoria de operación y la figura 1.7 muestra los perfiles de trayectoria. Cabe recordar que en este trabajo de tesis se modelará y programará una trayectoria espacial de configuración tipo helicoidal.



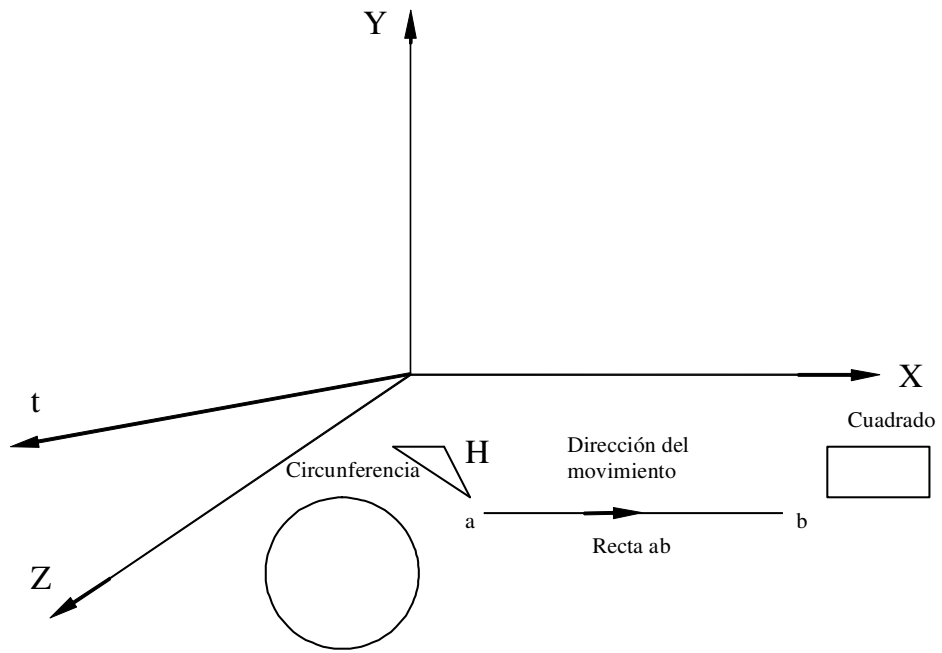


Figura 1.6 Lugares geométricos de trayectoria

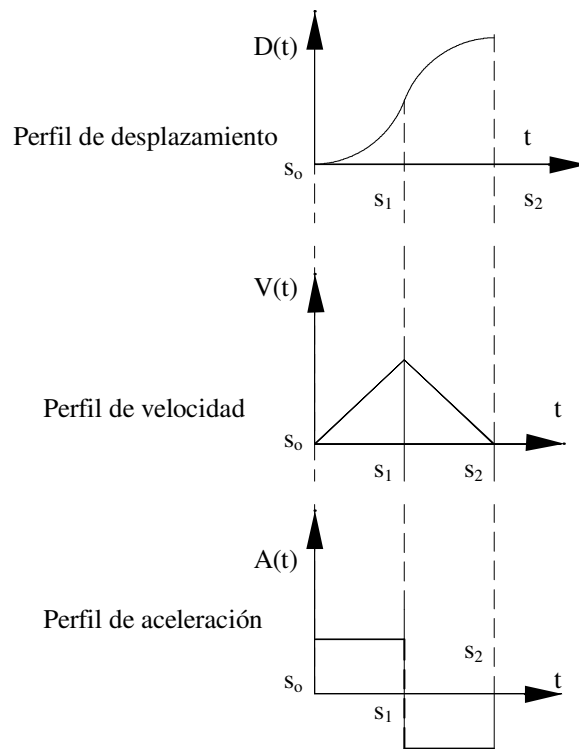


Figura 1.7 Perfiles de trayectoria

Finalmente, en la modelación de robots se deben formular, una vez obtenidos los modelos cinemáticos, dos problemas: el problema inverso y el problema directo. El primer problema consiste en determinar los parámetros de los actuadores dada una trayectoria. Esto es, la formulación del problema debe dar como solución los parámetros necesarios para mover al robot dada una trayectoria específica. El problema directo es al revés; esto es, si se conocen los parámetros de los actuadores el modelo debe ser capaz de localizar las posiciones y orientaciones del órgano terminal. En robótica tanto el problema directo como el inverso deben ser formulados con modelos de sistemas de ecuaciones cuadradas.

### **1.3 Resumen de capitulado**

En este trabajo de tesis se modelará la cinemática de posicionamiento de un robot industrial PUMA 560 usando el álgebra de Cuaterniones. La tesis aquí presentada, se divide en 5 capítulos los cuales se resumen a continuación:

En el capítulo 1 se presenta el motivo de la investigación, la definición de robot y sus elementos.

En el capítulo 2 se presenta la definición del problema y sus restricciones, así como una hipótesis y algunas consideraciones importantes.

En el capítulo 3 se presenta un breve resumen del álgebra de Cuaterniones y algunas de sus propiedades fundamentales, así como la representación paramétrica de rotaciones.

En el capítulo 4 se presenta un método para modelar la composición de rotaciones. Dicho método permitirá simplificar las operaciones algebraicas en el cálculo de las operaciones.

En el capítulo 5 se presenta el modelo cinemático del robot estudiado en este trabajo. Son formulados los problemas cinemáticos inverso y directo, respectivamente. Se presenta los bucles de programación de los modelos hechos en Mathematica [12].

Finalmente se presentan las conclusiones de esta tesis.

## Capítulo 2

### Definición del problema, restricciones e hipótesis

**Introducción.** En este capítulo se presenta la definición formal del problema por solucionar en esta tesis, sus restricciones fundamentales, así como una hipótesis y algunas consideraciones fundamentales.

#### 2.1 El problema y las restricciones

El problema por resolver en este trabajo de tesis se define de la manera siguiente:

*“Se requiere modelar, programar y simular el comportamiento cinemático de una cadena cinemática”.*

Considérese las siguientes restricciones:

- La cadena cinemática tiene 6 grados de libertad ( $R_1$ ).
- La cadena simula la arquitectura del robot industrial PUMA 560 ( $R_2$ ).
- Los eslabones del robot son rígidos ( $R_3$ ).
- El movimiento del robot es en el espacio ( $R_4$ ).
- El elemento terminal tiene como trayectoria una curva helicoidal dada por una función paramétrica ( $R_5$ ).
- No se incluyen en la modelación los perfiles de trayectoria ( $R_6$ ).
- Solo se modela el desplazamiento ( $R_7$ ).
- Las dimensiones principales del sistema mecánico son conocidas ( $R_8$ ).
- Las uniones son del tipo rotacional ( $R_9$ ).
- El modelo debe realizarse con el álgebra de Cuaterniones ( $R_{10}$ ).
- La programación debe ser en Mathematica©( $R_{11}$ ).

El problema descrito anteriormente se puede sintetizar de la manera siguiente:

*“Encontrar el modelo cinemático de posicionamiento de un sistema de multicuerpos rígidos acoplados de 6 grados de libertad (PUMA) usando el álgebra de Cuaterniones que permita formular los problemas directos e inversos tal que:*

*Las restricciones  $R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10}$  y  $R_{11}$*

*sean satisfechas”.*

La figura siguiente muestra la arquitectura del robot industrial PUMA 560 y una trayectoria helicoidal.

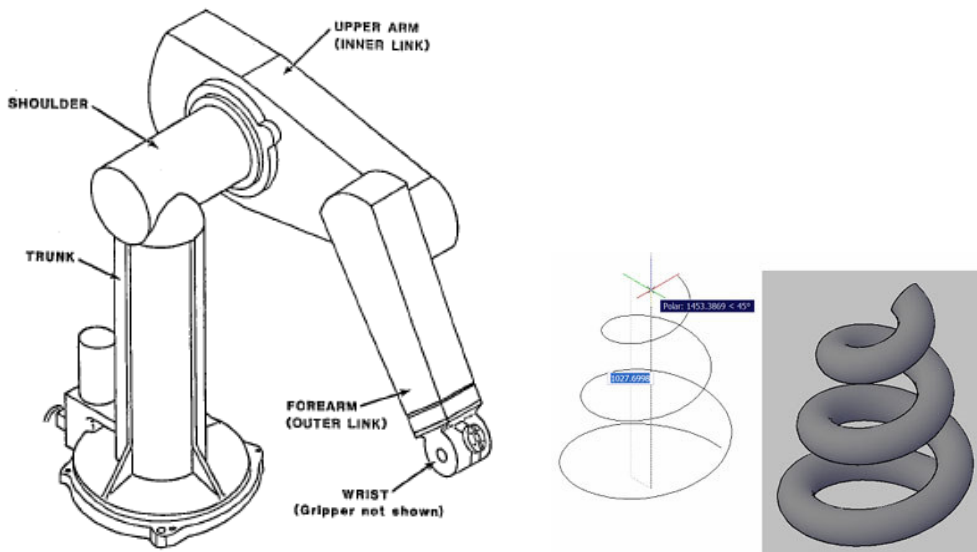


Figura 1.1 Configuración de robot y trayectoria

## **2.2 Hipótesis**

Todo problema planteado, exige una hipótesis. En este trabajo de tesis se usará la siguiente:

Considere la siguiente hipótesis:

*“La cinemática es la geometría del movimiento, por lo tanto, las premisas básicas asociadas con la hipótesis son las de la cinemática”.*

## **2.3 Algunas consideraciones importantes**

En este trabajo de tesis, se analizarán dos configuraciones relacionadas con el robot motivo de estudio. La primera de dichas configuraciones es llamada “no deformada” y la segunda “deformada” [5]. Tales configuraciones son también conocidas como configuración inicial y configuración final, respectivamente.

Por otro lado, no son consideradas las funciones de análisis en términos del tiempo por lo que las configuraciones no deformada y deformada son fijas en el análisis y solo se modelan los cambios de base en funciones de las rotaciones de los eslabones del robot de una configuración a otra.

## Capítulo 3

### El marco teórico

**Introducción.** En este capítulo se describe, en forma general, el álgebra de Cuaterniones [1], así como algunas de sus propiedades fundamentales, las cuales serán utilizadas en la modelación y programación del robot motivo de estudio en este trabajo de tesis. Se presenta, además, un método que será utilizado como guía para la modelación del robot.

#### 3.1 El álgebra de Cuaterniones

Los Cuaterniones fueron utilizados en años anteriores, entre otras cosas, para representar la forma paramétrica de rotaciones finitas de cuerpos rígidos o deformables, basándose en representaciones puramente geométricas, presentándolos como una generalización de los números complejos. Sin embargo, en sus aplicaciones no se encuentran las definiciones precisas y apropiadas de sus conceptos. Considérese el siguiente desarrollo:

Sea el conjunto en  $\mathfrak{R}^4$  en el cual se definen dos operaciones binarias  $\oplus : \mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$  y  $*$  :  $\mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$ . Esto es,

- i)  $(a,b,c,d) \oplus (\alpha,\beta,\gamma,\delta) = (a + \alpha, b + \beta, c + \gamma, d + \delta) ; \forall (a,b,c,d), (\alpha,\beta,\gamma,\delta) \in \mathfrak{R}^4$
- ii)  $(a,b,c,d) * (\alpha,\beta,\gamma,\delta) = (a\alpha - b\beta - c\gamma - d\delta, a\beta + b\alpha + c\delta - d\gamma, a\gamma + b\delta + c\alpha + d\beta, a\delta + b\gamma - c\beta + d\alpha), \forall (a,b,c,d), (\alpha,\beta,\gamma,\delta) \in \mathfrak{R}^4$ .

La operación  $\oplus : \mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$  es la suma usual en  $\mathfrak{R}^4$  y es bien conocido que el conjunto  $(\mathfrak{R}^4, \oplus)$  es un grupo aditivo conmutativo y que la terna  $(\mathfrak{R}^4, \oplus, *)$  es un cuerpo no conmutativo [1]. Por otro lado, la operación  $\bullet : \mathfrak{R} \times \mathfrak{R}^4 \rightarrow \mathfrak{R}^4$  definida por:

$$\alpha \bullet (a,b,c,d) = (a\alpha, b\alpha, c\alpha, d\alpha), \forall (a,b,c,d) \in \mathfrak{R}^4, \quad \alpha \in \mathfrak{R}$$

es una multiplicación escalar en  $\mathfrak{R}^4$ . Por lo tanto, la terna  $(\mathfrak{R}^4, \oplus, \bullet)$  es un espacio vectorial real. La transformación  $\langle \bullet, \bullet \rangle : \mathfrak{R}^4 \times \mathfrak{R}^4 \rightarrow \mathfrak{R}$ , dada por:

$$\langle \mathbf{p}, \mathbf{q} \rangle = \sum_{i=0}^3 p_i q_i \in \mathfrak{R}$$

es un producto interno en  $\mathfrak{R}^4$  y la norma asociada es:

$$\|\mathbf{p}\| = \langle \mathbf{p}, \mathbf{p} \rangle^{\frac{1}{2}} = (p_0^2 + p_1^2 + p_2^2 + p_3^2)^{\frac{1}{2}}$$

Por lo tanto, la estructura  $\mathcal{Q}=(\mathfrak{R}^4, \oplus, *, \bullet, \|\bullet\|)$ , es un espacio vectorial normado el cual se llamará Espacio vectorial de Cuaterniones y sus elementos Cuaterniones [1].

Considérese ahora los siguientes subespacios vectoriales de  $\mathcal{Q}$ :

$$Q_R = (a, 0, 0, 0) : a \in \mathfrak{R} \subset \mathcal{Q}; \quad Q_V = ((0, b, c, d) : b, c, d \in \mathfrak{R}^3),$$

donde  $Q_R$  y  $Q_V$  es un espacio vectorial real en  $\mathfrak{R}$  y  $Q_V$  es un espacio en  $\mathfrak{R}^3$ , luego entonces se puede representar el espacio  $\mathcal{Q}$  como sigue [1]:

$$\mathcal{Q} = Q_R \oplus Q_V \equiv \{ \mathbf{p} \in \mathcal{Q} : \mathbf{p} = p_R \oplus p_V, p_R \in Q_R, p_V \in Q_V \}.$$

Las transformaciones  $T_R: Q_R \rightarrow \mathfrak{R}$  y  $T_V: Q_V \rightarrow \mathfrak{R}^3$ , definidas por :

$$T_R(a, 0, 0, 0) = a \quad \text{y} \quad T_V(0, b, c, d) = (b, c, d)$$

son isomorfismos. Por lo tanto, si  $\mathbf{p}=(a,b,c,d) \in \mathcal{Q}$ , entonces se puede expresar a “ $\mathbf{p}$ ” como sigue:

$$\mathbf{p} = T_R^{-1}(a) \oplus T_V^{-1}(b, c, d).$$

Por otro lado, un Cuaternión conjugado  $\bar{\mathbf{p}} \in \mathcal{Q}$  de  $\mathbf{p} = (a, b, c, d)$  está definido por:

$$\bar{\mathbf{p}} = (a, -b, -c, -d).$$

Las propiedades siguientes se satisfacen [1]:

- 1)  $\overline{p \oplus q} = \overline{p} \oplus \overline{q}$
- 2)  $\overline{p * q} = \overline{q} * \overline{p}$
- 3)  $p * \overline{p} = \overline{p} * p \in Q_R$

### 3.2 Representación paramétrica de las rotaciones

Sea  $\rho(p, \bullet) : Q \rightarrow Q, p \in Q$ , entonces:

$$\rho(p, q) = p * q * p^{-1} = \frac{1}{\|p\|^2} \bullet (p * q * \overline{p}), \quad \forall p, q \in Q$$

Dicha transformación lineal es una rotación la cual preserva el producto interno, la norma y el ángulo [9]. Por otro lado, la matriz  $M_{\rho(p, \bullet)} \in M_{4 \times 4}$  asociada con la transformación  $\rho(p, \bullet) \in L(Q, Q)$ , dada la base  $B = \{e_j\}_{j=1}^4$  es :

$$M_{\rho(p, \bullet)} = \frac{1}{\|p\|^2} \begin{pmatrix} \|p\|^2 & 0 & 0 & 0 \\ 0 & p_0^2 + p_1^2 - p_2^2 - p_3^2 & 2(p_1 p_2 - p_3 p_0) & 2(p_0 p_2 + p_1 p_3) \\ 0 & 2(p_0 p_3 + p_1 p_2) & p_0^2 - p_1^2 + p_2^2 - p_3^2 & 2(p_2 p_3 - p_0 p_1) \\ 0 & 2(p_1 p_3 - p_0 p_2) & 2(p_0 p_1 + p_2 p_3) & p_0^2 - p_1^2 - p_2^2 + p_3^2 \end{pmatrix}$$

siendo  $p = (p_0, p_1, p_2, p_3) \in Q$ . Finalmente, las relaciones geométricas entre los Cuaterniones y los componentes de las rotaciones son las siguientes [1]:

$$p_0 = \|p\| \cos \frac{\theta}{2}, \quad p_V = \pm \|p\| \sin \frac{\theta}{2} \underline{\omega}$$

Aquí,  $\theta \in \mathfrak{R}$  es el ángulo de la rotación y  $\underline{\omega} \in \mathfrak{R}^3$  es el eje de la rotación.



### 3.3 Método de modelación

En esta sección se presenta un método de modelación el cual será de utilidad para construir los modelos cinemáticos relacionados con el robot motivo de estudio en este trabajo de tesis. Considérense los siguientes pasos [13]:

- 1) Caracterizar el robot.
- 2) Construir las ecuaciones de posición en la configuración de referencia usando la rotación de Cuaterniones para la formulación del problema inverso.
- 3) Formular el problema cinemático inverso en la configuración no deformada.
- 4) Construir las ecuaciones de posición en la configuración de referencia usando la rotación de Cuaterniones para la formulación del problema directo.
- 5) Formular el problema cinemático directo en la configuración no deformada.
- 6) Construir las ecuaciones de posición en la configuración deformada usando la rotación de Cuaterniones para la formulación del problema inverso.
- 7) Formular el problema cinemático inverso en la configuración deformada.
- 8) Construir las ecuaciones de posición en la configuración deformada usando la rotación de Cuaterniones para la formulación del problema directo.
- 9) Formular el problema cinemático directo en la configuración deformada.
- 10) Programar y simular las ecuaciones generadas en Mathematica.

## Capítulo 4

### Algunas consideraciones sobre la composición de rotaciones

**Introducción.** El objetivo en este capítulo, es describir, con la ayuda de las propiedades del álgebra de Cuaterniones [1], un método que permita describir de forma sistemática la composición de rotaciones. Para tal propósito se usará un robot de 2 GDL.

#### 4.1 Las rotaciones entre sistemas de referencia no inercial y móvil

Considérense un robot compuesto por dos eslabones rígidos como se muestra en la figura 4.1. Sobre los eslabones que componen al robot se localizan sistemas locales, uno definido entre el primer eslabón y la base fija y otro en la intersección de los eslabones móviles. Tales sistemas ortogonales son llamados bases móviles. Por otro lado, de acuerdo con la figura 4.1, sobre el sistema de coordenadas se asocia una base referencial fija o también llamada inercial.

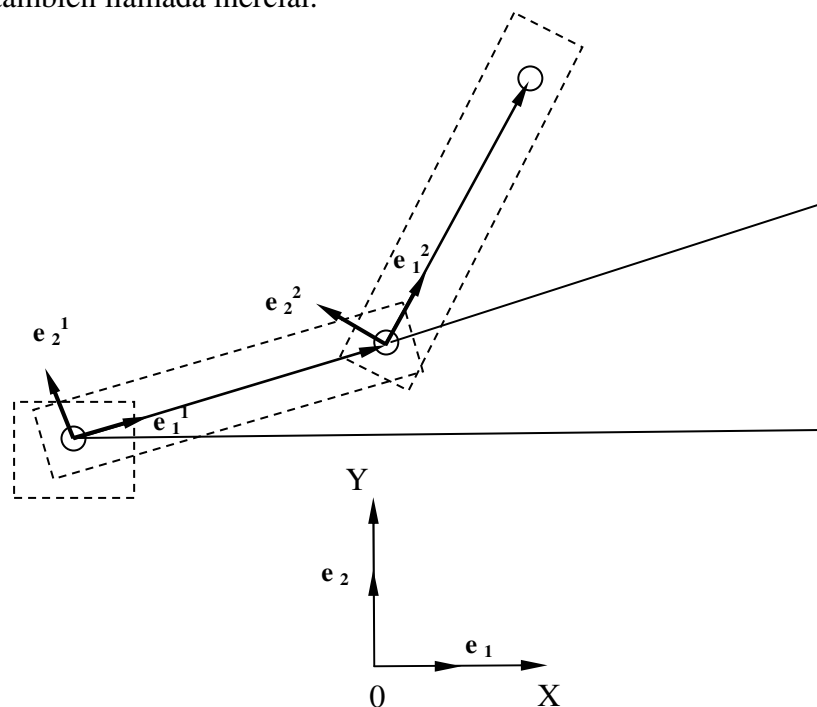


Figura 4.1 Robot de GDL y bases

Considérese la siguiente proposición:

“Las rotaciones de la base inercial fija  $e_j$ ,  $j=1,2,3$  sobre las bases móviles  $e_j^1$  y  $e_j^2$  se modelan con la función:

$$\rho(p, q) = p * q * p^{-1} = \frac{1}{\|p\|^2} \bullet (p * q * \bar{p}), \quad \forall p, q \in Q ”$$

Por lo tanto, la rotación de la base inercial para obtener la primera base móvil se modela de la manera siguiente:

$$e_j^1 = \rho(p^1, e_j) = p^1 * e_j * \bar{p}^1 \quad (4.1)$$

Siendo “ $p^1$ ” un Cuaternión unitario relacionado con la rotación de la base canónica  $e_j$  sobre la base  $e_j^1$ ; esto es:

$$\|p^1\| = 1$$

Considere ahora la base  $e_j^2$ . La rotación de la base  $e_j^1$  para obtener la  $e_j^2$  se modela del siguiente modo:

$$e_j^2 = \rho(p^2, e_j^1) \quad (4.2)$$

O, en forma explícita,

$$e_j^2 = \rho(p^2, \rho(p^1, e_j)) = p^2 * p^1 * e_j * \overline{p^1 * p^2} \quad (4.3)$$

## 4.2 Las rotaciones entre las configuraciones

Considérese que el robot ha sido movido de su posición original a una configuración deformada como se muestra en la figura 4.2.

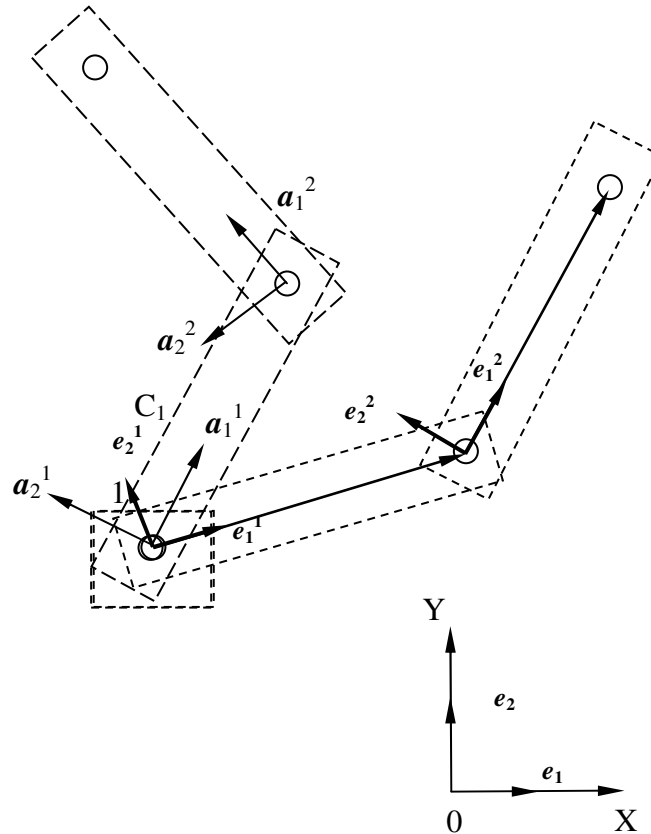


Figura 4.2 Bases deformadas

La bases  $a_j^1$  y  $a_j^2$  son rotaciones de las bases  $e_j^1$  y  $e_j^2$ . Dichas bases se representan en términos de Cuaterniones de la manera siguiente:

$$\begin{aligned}
 1) \quad a_j^1 &= \rho(P^1, e_j^1) = P^1 * e_j^1 * \overline{P^1} \\
 2) \quad a_j^2 &= \rho(P^2, a_j^1) = \rho(P^2, \rho(P^1, e_j^1)) = P^2 * P^1 * e_j^1 * \overline{P^1 * P^2}
 \end{aligned}
 \tag{4.4}$$

### 4.3 Generalización de la composición de rotaciones

Sea  $e_j$ ,  $j=1,2,3$  un sistema fijo y  $e_j^1$ ,  $j=1,2,3$  el primer sistema ortonormal afectado por una rotación definida por el Cuaternión  $p^1$ , de forma tal que:

$$e_j^1 = \rho(p^1, e_j) = p^1 * e_j * \overline{p^1} \quad (4.5)$$

Sea  $e_j^2$ ,  $j=1,2,3$  un segundo sistema afectado por la primera y por la segunda rotación, esta última definida por el Cuaternión  $p^2$ , de forma tal que:

$$e_j^2 = \rho(p^{12}, e_j^1) \quad (4.6)$$

Donde  $p^{12}$  es el Cuaternión  $p^2$  afectado por la primera rotación definida por  $p^1$ . Esto es:

$$p^{12} = \rho(p^1, p^2) = p^1 * p^2 * \overline{p^1} \quad (4.7)$$

Nótese que al sustituir la expresión (4.7) en la expresión (4.6) se obtiene:

$$e_j^2 = \rho(p^{12}, e_j^1) = \rho(p^1 * p^2 * \overline{p^1}, p^1 * e_j * \overline{p^1}) = p^1 * p^2 * \overline{p^1} * p^1 * e_j * \overline{p^1} * \overline{p^1 * p^2 * \overline{p^1}} \quad (4.8)$$

Considérense las propiedades siguientes,

$$1) \overline{\overline{p}} * p = 1 \quad (4.9)$$

$$2) \overline{p * q * p} = \overline{p} * \overline{p * q} \text{ y}$$

$$3) \overline{\overline{p}} = p$$

Al utilizar las expresiones (4.9) en la expresión (4.8) se obtiene que:

$$e_j^2 = p^1 * p^2 * 1 * e_j * \overline{p^1} * \overline{\overline{p^1}} * \overline{p^1 * p^2} = p^1 * p^2 * e_j * \overline{p^1} * p^1 * \overline{p^1 * p^2} \quad (4.10)$$

O, en forma equivalente:

$$e_j^2 = p^1 * p^2 * e_j * \overline{p^1 * p^2} = \rho(p^1 * p^2, e_j) \quad (4.11)$$

Este resultado se puede generalizar. Esto es:

$$e_j^n = \rho(p^1 * p^2 * \dots * p^n, e_j) \quad (4.12)$$

#### 4.4 Demostración de la composición de rotaciones

En esta sección se demuestra usando el software Mathematica® [12] la generalización de la composición de rotaciones. La tabla 4.1 muestra el código del programa.

<code>Rot[p_, q_] := p**q**p̄</code>	Definición de rotación. El doble * significa la no conmutatividad
<code>ej1=Rot[p,ej]; Print["ej1=",ej1]; ej1=p**ej**p̄</code>	ej es la base (j=1,2,3) ej1 es el primer sistema afectado por la rotación definida por “p”
<code>pq=Rot[p,q]; pr=Rot[p,r];</code>	“pq” es el Cuaternión “q” afectado por “p” pr es el Cuaternión “r” afectado por “p”
<code>ej2=Rot[pq,ej1]; Print["ej2=",ej2]; ej2=p**q**p̄**p**ej**p̄**p**q**p̄</code>	ej2 es la segunda base, afectada por “pq”
<code>qr=Rot[pq,pr];</code>	“qr” es el Cuaternión pq afectado por la rotación “pr”

<pre> ej3=Rot[qr,ej2]; Print["ej3=",ej3]; ej3=p**q**p̄**p**r**p̄**p**q**p̄**p**q**p̄**p**ej       **p̄**p**q**p̄**p**q**p̄**p**r**p̄**p**q**p̄ </pre>	<p>ej3 es el tercer base, afectada por el Cuaternión “qr”</p>
<pre> Unprotect[NonCommutativeMultiply] q_**p_ := p**q̄ p̄_ := p p̄_**p_ := 1 p_**1 := p </pre>	<p>Reglas para los Cuaterniones conjugados, definidos por (9)</p>
<pre> Print["ej1=",ej1]; Print["ej2=",ej2]; Print["ej3=",ej3]; ej1=p**ej**p̄ ej2=p**q**ej**q̄**p̄ ej3=p**q**r**ej**r̄**q̄**p̄ </pre>	<p>Al mostrarlos nuevamente se aplican las reglas anteriores y simplifican las ecuaciones de las bases</p>

Tabla 4.1 Código del programa para la composición de rotaciones

## Capítulo 5

### Modelación y simulación de la cinemática del robot PUMA 560

**Introducción.** En este capítulo serán construidos los modelos cinemáticos relacionados con el robot PUMA 560 motivo de estudio en este trabajo de tesis. Son formulados los problemas cinemáticos inverso y directo, respectivamente. La programación de los modelos generados será hecha en Mathematica [12, 15]

#### 5.1 Algunas consideraciones importantes para el desarrollo del modelo

La cinemática de un manipulador estudia el movimiento del dispositivo sin considerar las fuerzas que causan el movimiento. Las relaciones matemáticas entre el movimiento y las fuerzas son estudiadas en la dinámica. La cinemática es importante no solamente como un prerequisite indispensable para cualquier descripción dinámica, sino también para aplicaciones prácticas, tales como la planeación del movimiento y el análisis de singularidades.

En particular se estudia la relación entre las variables espaciales ligadas a las articulaciones (coordenadas articulares), y las variables que proporcionan la posición y orientación del extremo del manipulador (coordenadas operacionales).

La posición y orientación del órgano terminal de un robot son calculadas a partir de posiciones de las juntas por medio del modelo geométrico del brazo del robot. Para robots seriales, el mapeo de las posiciones de las juntas a la posición del efector final es relativamente sencillo. Por otro lado, el mapeo inverso es difícil, al menos en el caso de robots seriales que tienen una estructura cinemática arbitraria; esto es, la posición y orientación de los ejes de las juntas pueden ser cualquiera. Por lo tanto, muchos robots industriales tienen una ingeniería especialmente diseñada para reducir la complejidad matemática del mapeo inverso (con frecuencia con el costo de la complejidad mecánica incrementada).

Los problemas de la cinemática se dividen en dos: el problema cinemático directo y el problema cinemático inverso, el primero consiste en una vez dados los valores de las



articulaciones, determinar la posición y orientación del elemento terminal o efector, mientras que el segundo consiste en dada la posición y orientación del elemento terminal encontrar los valores correspondientes de las articulaciones que describen ese movimiento. Estos problemas se abordan individualmente y se plantean como objetivos de este trabajo de investigación.

## 5.2 Caracterización del multicuerpo en estudio

En esta sección se presenta, en forma general, la caracterización del robot PUMA motivo de estudio en este trabajo de tesis. La configuración del robot es mostrada en la figura 5.1

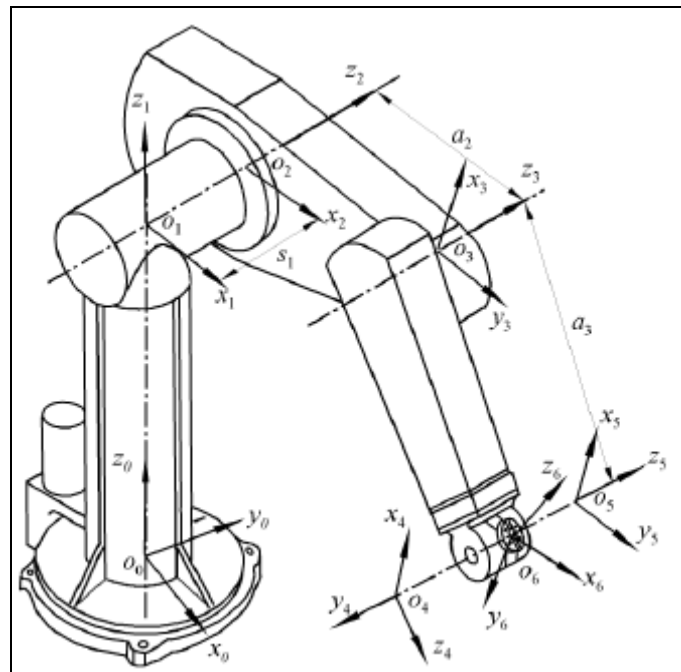


Figura 5.1 Configuración de un manipulador PUMA

El robot mostrado en la figura anterior, consta de 3 eslabones rígidos con los cuales se localiza la posición (X, Y, Z) y de un órgano terminal o efector el cual es caracterizado por 3 rotaciones que determinan la orientación. El robot posee juntas del tipo revolución (R), es de tipo serial y de 6 grados de libertad.

### 5.3 Modelación de la configuración no deformada

En esta sección se modela la configuración no deformada o de referencia asociada con el robot motivo de estudio. Dicha configuración se muestra en la figura 5.2. En la figura 5.3 se muestran las dimensiones principales del robot.

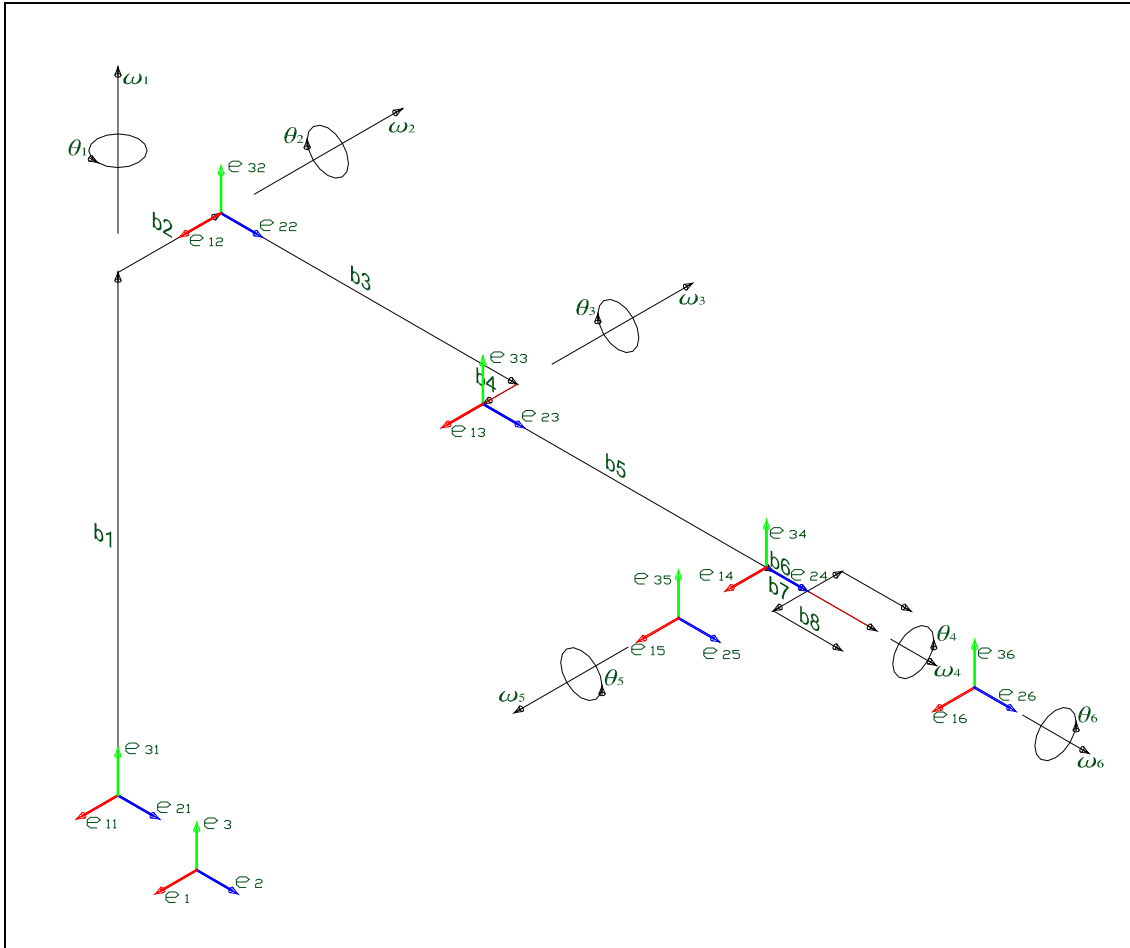


Figura 5.2 Bases y ejes de rotación del robot PUMA en la configuración no deformada

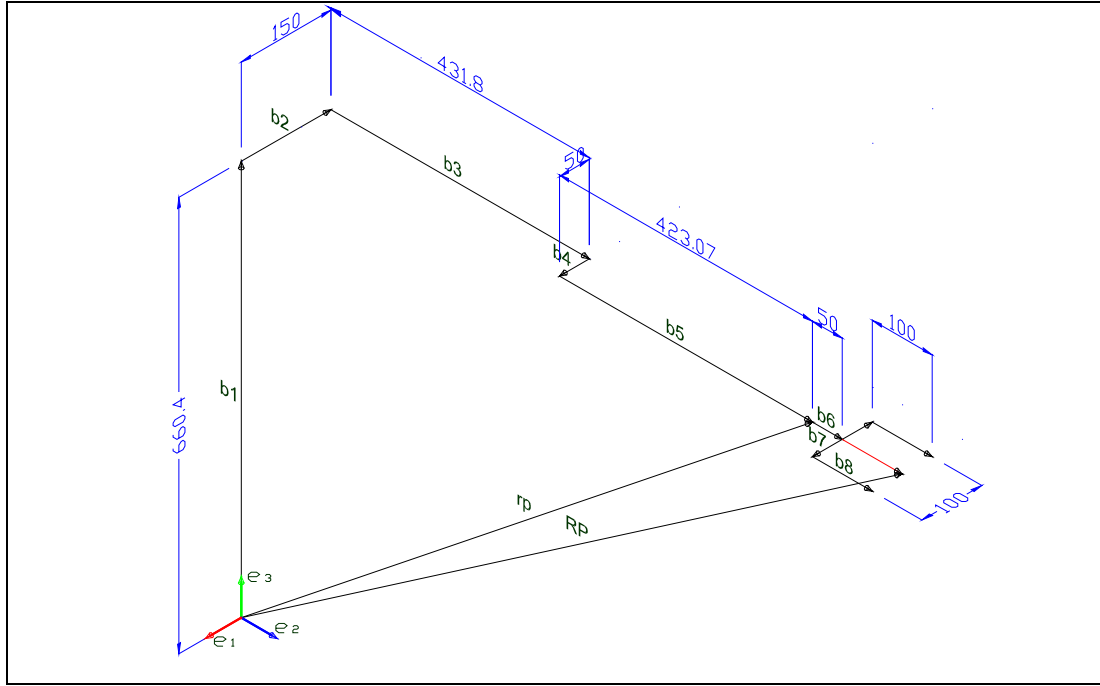


Figura 5.3 Dimensionamiento y vectores de posición

De acuerdo con la figura 5.2, cada eslabón tiene asociada una base local móvil referida al sistema local fijo  $e_j$  y cada eslabón tiene un eje de rotación representado por el símbolo “ $\omega$ ”. Por otro lado, las rotaciones de las bases móviles pueden representarse por medio del Cuaterniones unitarios; esto es:

$$\begin{aligned}
 e_j^1 &= \rho(P, e_j) \\
 e_j^2 &= \rho(P, \rho(Q, e_j)) \\
 e_j^3 &= \rho(P, \rho(Q, \rho(R, e_j))) \\
 e_j^4 &= \rho(P, \rho(Q, \rho(R, \rho(S, e_j)))) \\
 e_j^5 &= \rho(P, \rho(Q, \rho(R, \rho(S, \rho(T, e_j)))))) \\
 e_j^6 &= \rho(P, \rho(Q, \rho(R, \rho(S, \rho(T, \rho(U, e_j))))))
 \end{aligned}
 \quad j = 1, 2, 3 \quad (5.1)$$

Nótese que se ha usado la expresión (4.12) en la generación de las expresiones (5.1). Puesto que el Cuaternión  $P$  está relacionado con la rotación del primer eslabón, el Cuaternión  $Q$  con el segundo eslabón y así sucesivamente.

Por otro lado, los Cuaterniones  $P, Q, R, S, T$  y  $U$  tienen la forma siguiente:

$$\begin{aligned}
P &= \{P_0, P_1, P_2, P_3\} \\
Q &= \{Q_0, Q_1, Q_2, Q_3\} \\
R &= \{R_0, R_1, R_2, R_3\} \\
S &= \{S_0, S_1, S_2, S_3\} \\
T &= \{T_0, T_1, T_2, T_3\} \\
U &= \{U_0, U_1, U_2, U_3\}
\end{aligned} \tag{5.2}$$

Los ejes de rotación se pueden representar, según la figura 5.2, de la manera siguiente

$$\begin{aligned}
\omega_1 &= \{0,0,1\} \\
\omega_2 &= \{-1,0,0\} \\
\omega_3 &= \{-1,0,0\} \\
\omega_4 &= \{0,1,0\} \\
\omega_5 &= \{-1,0,0\} \\
\omega_6 &= \{0,1,0\}
\end{aligned} \tag{5.3}$$

El Cuaternión “ $P$ ” está relacionado con el eje de rotación “ $\omega_1$ ”, mostrado en la figura 5.2, de la siguiente forma:

$$P_0 = \text{Cos}(\theta_1 / 2); \quad P_v = \omega_1 \bullet \text{Sen}(\theta_1 / 2) \tag{5.4}$$

Nótese que si se sustituye “ $\omega_1$ ” de las expresiones (5.3) en “ $P_v$ ” se tiene que:

$$P_v = \{0,0,1\} \bullet \text{Sen}(\theta_1 / 2) = \{0,0, \text{Sen}(\theta_1 / 2)\} = \{0, 0, P_3\} \tag{5.5}$$

Por lo tanto,

$$1) P_0 = \text{Cos}(\theta_1 / 2) \tag{5.6}$$

$$2) P_3 = \pm \text{Sen}(\theta_1 / 2)$$

Y, en consecuencia, el Cuaternion “ $P$ ” se representa por:

$$P = \{P_0, 0, 0, P_3\} \tag{5.7}$$

De manera similar utilizando el segundo eje de rotación  $\omega_2$ , y el cuaternión correspondiente  $Q$  queda de la forma:

$Q_0 = \text{Cos}(\theta_1 / 2)$ ;  $Q_v = \omega_1 \bullet \text{Sen}(\theta_1 / 2)$ , y por lo tanto

$$Q = \{Q_0, Q_1, 0, 0\} \quad (5.8)$$

Los restantes Cuaterniones se representan:

$$\begin{aligned} R &= \{R_0, R_1, 0, 0\} \\ S &= \{S_0, 0, S_2, 0\} \\ T &= \{T_0, T_1, 0, 0\} \\ U &= \{U_0, 0, U_2, 0\} \end{aligned} \quad (5.9)$$

Por otro lado, de acuerdo con la figura 5.3, el vector **RP** que localiza el extremo final del robot desde el origen de referencia, se puede expresar de la manera siguiente:

$$RP = b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5 \oplus b_6 \quad (5.10)$$

Y el vector **rp** que determinan los primeros 3 grados de libertad, es decir sin considerar el elemento terminal como lo muestra la figura 5.3 esta determinado por:

$$rp = b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5$$

Cada vector asociado con los eslabones, se puede representar en términos de las bases locales. Esto es:

$$\begin{aligned} b_1 &= d_1 e_3^1 \\ b_2 &= d_2 (-e_1^1) \\ b_3 &= d_3 e_2^2 \\ b_4 &= d_4 (-e_1^3) \\ b_5 &= d_5 e_2^4 \\ b_6 &= d_6 e_2^6 \end{aligned} \quad (5.11)$$

Aquí,  $d_1, d_2, d_3, d_4, d_5$  y  $d_6$  son las dimensiones principales del robot motivo de estudio. Por otro lado, las equivalencias siguientes:

$$e_j^6 = e_j^5 = e_j^4 = e_j^3 = e_j^2 = e_j^1 = e_j, \quad j = 1, 2, 3. \quad (5.12)$$

se satisfacen si:

$$P = \{1,0,0,0\} \quad (5.13)$$

$$Q = \{1,0,0,0\}$$

$$R = \{1,0,0,0\}$$

$$S = \{1,0,0,0\}$$

$$T = \{1,0,0,0\}$$

$$U = \{1,0,0,0\}$$

Las condiciones anteriores están relacionadas con una configuración no deformada conocida.

#### 5.4 Modelación de la configuración deformada

En esta sección se presenta el modelado de la configuración deformada relacionada con el robot motivo de estudio en esta tesis. Dicha modelación está asociada con el problema cinemático directo. Considere que el robot motivo de estudio ha sido movido de una configuración inicial a una final llamada deformada. Los sistemas locales relacionados con la configuración deformada se expresan de la manera siguiente:

$$\begin{aligned}
 e_j^1 &= \rho(\mathbf{p}, \rho(\mathbf{P}, e_j)) \\
 e_j^2 &= \rho(\mathbf{p}, \rho(\mathbf{P}, \rho(\mathbf{q}, \rho(\mathbf{Q}, e_j)))) \\
 e_j^3 &= \rho(\mathbf{p}, \rho(\mathbf{P}, \rho(\mathbf{q}, \rho(\mathbf{Q}, \rho(\mathbf{r}, \rho(\mathbf{R}, e_j))))) \\
 e_j^4 &= \rho(\mathbf{p}, \rho(\mathbf{P}, \rho(\mathbf{q}, \rho(\mathbf{Q}, \rho(\mathbf{r}, \rho(\mathbf{R}, \rho(\mathbf{s}, \rho(\mathbf{S}, e_j))))))) \quad j = 1, 2, 3 \\
 e_j^5 &= \rho(\mathbf{p}, \rho(\mathbf{P}, \rho(\mathbf{q}, \rho(\mathbf{Q}, \rho(\mathbf{r}, \rho(\mathbf{R}, \rho(\mathbf{s}, \rho(\mathbf{S}, \rho(\mathbf{t}, \rho(\mathbf{T}, e_j)))))))) \\
 e_j^6 &= \rho(\mathbf{p}, \rho(\mathbf{P}, \rho(\mathbf{q}, \rho(\mathbf{Q}, \rho(\mathbf{r}, \rho(\mathbf{R}, \rho(\mathbf{s}, \rho(\mathbf{S}, \rho(\mathbf{t}, \rho(\mathbf{T}, \rho(\mathbf{u}, \rho(\mathbf{U}, e_j))))))))))
 \end{aligned} \quad (5.14)$$

Siendo  $\mathbf{p}$ ,  $\mathbf{q}$ ,  $\mathbf{r}$ ,  $\mathbf{s}$ ,  $\mathbf{t}$  y  $\mathbf{u}$  los Cuaterniones relacionados con las rotaciones de las bases de la configuración no deformada a la deformada. Considérense que las relaciones siguientes se satisfacen:

$$P = Q = R = S = T = U = I \quad (5.15)$$

Siendo  $I$  el Cuaternión representativo del elemento neutro de la multiplicación, el cual es definido por:

$$I = \{1,0,0,0\}$$

De acuerdo con las consideraciones (5.15) las expresiones (5.14) se reducen a:

$$\begin{aligned}
e_j^1 &= \rho(\varphi, e_j) \\
e_j^2 &= \rho(\varphi, \rho(\hat{q}, e_j)) \\
e_j^3 &= \rho(\varphi, \rho(\hat{q}, \rho(\hat{t}, e_j))) \\
e_j^4 &= \rho(\varphi, \rho(\hat{q}, \rho(\hat{t}, \rho(\hat{s}, e_j)))) \\
e_j^5 &= \rho(\varphi, \rho(\hat{q}, \rho(\hat{t}, \rho(\hat{s}, \rho(\hat{t}, e_j)))) \\
e_j^6 &= \rho(\varphi, \rho(\hat{q}, \rho(\hat{t}, \rho(\hat{s}, \rho(\hat{t}, \rho(\hat{u}, e_j))))))
\end{aligned} \quad j = 1,2,3 \quad (5.16)$$

La forma explícita de las relaciones (5.16) es:

$$\begin{aligned}
e_j^1 &= p^* e_j^* \bar{p} \\
e_j^2 &= p^* q^* e_j^* \overline{p^* q} \\
e_j^3 &= p^* q^* r^* e_j^* \overline{p^* q^* r} \\
e_j^4 &= p^* q^* r^* s^* e_j^* \overline{p^* q^* r^* s} \\
e_j^5 &= p^* q^* r^* s^* t^* e_j^* \overline{p^* q^* r^* s^* t} \\
e_j^6 &= p^* q^* r^* s^* t^* u^* e_j^* \overline{p^* q^* r^* s^* t^* u}
\end{aligned} \quad (5.17)$$

Por lo tanto, la expresión (5.10) se escribe en términos de las expresiones (5.11), (5.13), (5.14) y (5.17) de la manera siguiente:

$$\begin{aligned}
RP &= d_1(p^* e_j^* \bar{p}) \oplus d_2(p^* q^* e_j^* \overline{p^* q}) \oplus d_3(p^* q^* r^* e_j^* \overline{p^* q^* r}) \\
&\oplus d_4(p^* q^* r^* s^* e_j^* \overline{p^* q^* r^* s}) \oplus d_5(p^* q^* r^* s^* t^* e_j^* \overline{p^* q^* r^* s^* t}) \\
&\oplus d_6(p^* q^* r^* s^* t^* u^* e_j^* \overline{p^* q^* r^* s^* t^* u})
\end{aligned} \quad (5.18)$$

## 5.5 Cinemática directa

En esta sección se formula el problema cinemática directo relacionado con el robot PUMA 560 motivo de estudio en este trabajo de tesis.

### 5.5.1 Problema cinemático directo

*Dados  $d_1, d_2, d_3, d_4, d_5, d_6 \in \mathfrak{R}$ ,  $\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s}, \mathbf{t}, \mathbf{u} \in \mathfrak{R}^4$ , con  $\|\mathbf{p}\| = 1$ ,  $\|\mathbf{q}\| = 1$ ,  $\|\mathbf{r}\| = 1$ ,  $\|\mathbf{s}\| = 1$ ,  $\|\mathbf{t}\| = 1$ , y  $\|\mathbf{u}\| = 1$ , encuentre  $\mathbf{RP} \in \mathfrak{R}^3$  tal que la expresión (4.18) sea satisfecha”*

### 5.5.2 Simulación de la cinemática directa

El modelo descrito en la sección anterior, se puede resumirse de la manera siguiente:

*Dados los valores de los Cuaterniones  $\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s}, \mathbf{t}$  y  $\mathbf{u}$  encontrar la trayectoria realizada por el elemento terminal del manipulador.*

A continuación se muestran algunas funciones generales utilizadas para realizar la simulación del manipulador: Con el objetivo de que se muestre de forma similar a un manipulador real, se tiene una función denominada “Elemento”, esto con el fin de mostrar de forma gráfica un cubo orientado en los ejes  $\mathbf{e}_1$ ,  $\mathbf{e}_2$  y  $\mathbf{e}_3$ , de longitud  $l$  en dirección  $i$ . La figura 5.4 muestra algunos puntos de interés para graficar en Mathematica y la figura 5.5 muestra la caracterización del órgano terminal.



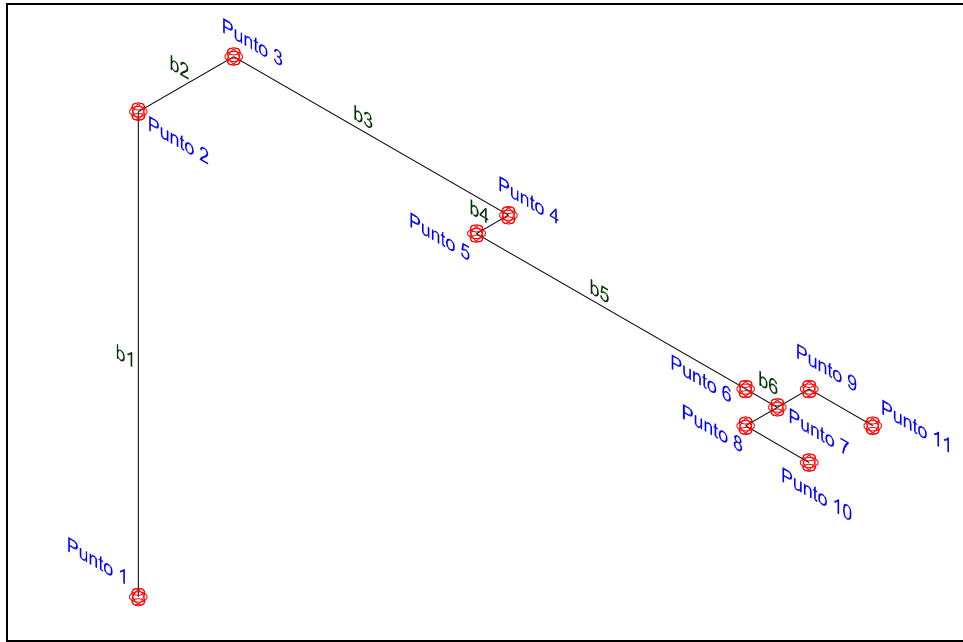


Figura 5.4 Puntos de referencia e cada elemento

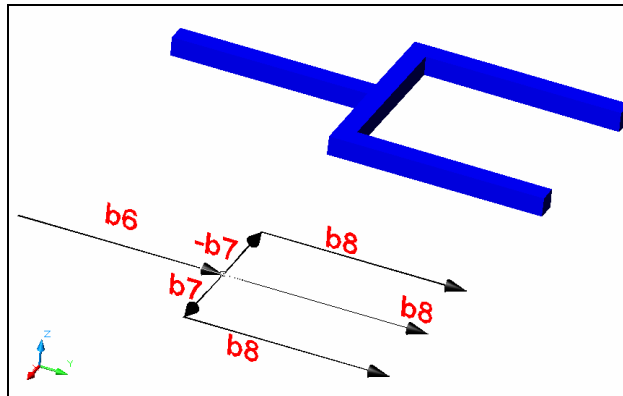


Figura 5.5 Órgano terminal

Por otro lado, la orientación del efector final debe ser especificada por tres variables independientes. Los vectores  $n_1$ ,  $n_2$  y  $n_3$  son representados por nueve componentes en el sistema cartesiano base coordenado. Sin embargo 2 de estos 3 vectores determinan el tercer vector unitario, porque los vectores representan un marco coordenado ortogonal de mano derecha. Por otra parte la ortogonalidad y la longitud de los dos vectores restantes imponen tres relaciones adicionales reduciendo el número de variables independientes que describen la orientación a tres. La tabla 5.1 muestra la programación hecha en Mathematica [12] del modelo de la cinemática directa.

Funciones generales	
<pre> Elemento[q1_,q2_,l_,i_,e1_,e2_,e3_] :=Module[ {p1,p2,p3,p4,p5,p6,p7,p8},   Switch[i,     0,     p1=q1+(-1/2)*e2+(-1/2)*e3;     p2=q1+1/2*e2+(-1/2)*e3;     p3=q1+1/2*e2+1/2*e3;     p4=q1+(-1/2)*e2+1/2*e3;     p5=q2+(-1/2)*e2+(-1/2)*e3;     p6=q2+1/2*e2+(-1/2)*e3;     p7=q2+1/2*e2+1/2*e3;     p8=q2+(-1/2)*e2+1/2*e3,     1,     p1=q1+(1/2)*e1+(-1/2)*e3;     p2=q1+(-1/2)*e1+(-1/2)*e3;     p3=q1+(-1/2)*e1+(1/2)*e3;     p4=q1+(+1/2)*e1+(1/2)*e3;     p5=q2+(+1/2)*e1+(-1/2)*e3;     p6=q2+(-1/2)*e1+(-1/2)*e3;     p7=q2+(-1/2)*e1+(1/2)*e3;     p8=q2+(+1/2)*e1+(1/2)*e3,     2,     p1=q1+(+1/2)*e1+(-1/2)*e2;     p2=q1+(+1/2)*e1+(+1/2)*e2;     p3=q1+(-1/2)*e1+(+1/2)*e2;     p4=q1+(-1/2)*e1+(-1/2)*e2;     p5=q2+(+1/2)*e1+(-1/2)*e2;     p6=q2+(+1/2)*e1+(+1/2)*e2;     p7=q2+(-1/2)*e1+(+1/2)*e2;     p8=q2+(-1/2)*e1+(-1/2)*e2]; Graphics3D[{   {RGBColor[1,1,0],Polygon[{p1,p2,p3,p4}]},   {RGBColor[1,0,1],Polygon[{p1,p4,p8,p5}]},   {RGBColor[0,1,1],Polygon[{p5,p6,p7,p8}]},   {RGBColor[1,0,0],Polygon[{p6,p2,p3,p7}]},   {RGBColor[0,0,1],Polygon[{p8,p7,p3,p4}]},   {RGBColor[0,1,0],Polygon[{p6,p2,p1,p5}]}, {Line[{q1,q1+50*e1}]}, {Line[{q1,q1+50*e2}]}, {Line[{q1,q1+50*e3}]}}, AspectRatio-&gt;Automatic]; </pre>	<p>Regresa un elemento gráfico orientado en el eje <math>i</math> de lado <math>l</math> y longitud <math>q2-q1</math></p>
<pre> ProductoQ[P_Quaternion,Q_Quaternion] := Module[{p=P,q=Q},p**q]; </pre>	<p>Producto entre Cuaterniones,</p>

	el doble * indica que es no conmutativa.
RotacionQ[P_Quaternion,Q_Quaternion]:= Module[{p=P,q=Q},p**q**Conjugate[p]];	Rotación definida con cuaterniones.
toQuaternion[t_]:=Quaternion[0,t[[1]],t[[2]],t[[3]]];	Función de transformación de $\mathbb{R}^3 \rightarrow \mathbb{R}^4$ , De Vector a Cuaternión.
fromQuaternion[q_Quaternion]:={q[[2]],q[[3]],q[[4]]};	Función de transformación de $\mathbb{R}^4 \rightarrow \mathbb{R}^3$ , De Cuaternión a Vector.
Rotacion[p1_,p2_]:= Module[{P1=p1,P2=p2,q1,q2}, If[Dimensions[P1]==={4}, q1=P1, q1=toQuaternion[P1]]; If[Dimensions[P2]==={4}, q2=P2, q2=toQuaternion[P2]]; fromQuaternion[RotacionQ[q1,q2]] ];	Función que utiliza la rotación entre Cuaterniones previamente definida.
TangenteF[f_,v_]:= Module[{l,F}, F=D[f,v]; l=(F[[1]]^2+F[[2]]^2+F[[3]]^2)^(1/2); Simplify[F/l] ];	Tangente unitaria a la función $f[v]$ .
NormalF[f_,v_]:= Module[{l,F}, F=D[TangenteF[f,v],v]; {l=(F[[1]]^2+F[[2]]^2+F[[3]]^2)^(1/2)}; Simplify[F/l] ];	Normal unitaria a la función $f[v]$ .
TxN[f1_,f2_]:= Module[ {l1=(f1[[1]]^2+f1[[2]]^2+f1[[3]]^2)^(1/2), l2=(f2[[1]]^2+f2[[2]]^2+f2[[3]]^2)^(1/2),F1,F2},	Producto Cruz entre dos vectores

<pre>F1=f1/l1; F2=f2/l2; Simplify[Cross[F1,F2]] ];</pre>	
<pre>Norma[v_]:=Sqrt[v[[1]]^2+v[[2]]^2+v[[3]]^2];</pre>	<p>Norma de un vector</p>
<pre>Cilindro[r_,H_,o_,q_,v_]:= Module[ {qq=q,i,h,n,l,l1,f,ff,g,v1,vv, rsphi,rcphi,rsphi2,rcphi2,mesh,pp1,pp2,pp3,pp}, If[o&lt;0,h=-H,h=H]; Switch[Abs[o], 1,ff={h,r Cos[2Pi i/n],r Sin[2Pi i/n]},(* x *) 2,ff={r Cos[2Pi i/n],h,r Sin[2Pi i/n]},(* y *) 3,ff={r Cos[2Pi i/n],r Sin[2Pi i/n],h} (* z *) ]; vv = fromQuaternion[ RotacionQ[qq,toQuaternion[v]] ]; f=vv+ fromQuaternion[RotacionQ[qq,toQuaternion[ff]]]; g=vv+fromQuaternion[ RotacionQ[qq,toQuaternion[ff/.h-&gt;0]] ]; n=10; v1=Table[ rcphi=N[f[[1]]]; rsphi=N[f[[2]]]; rcphi2=N[g[[1]]]; rsphi2=N[g[[2]]]; {{rcphi,rsphi,f[[3]]}, {rcphi2,rsphi2,g[[3]]}},{i,n} ]; l=v1; l1=RotateLeft/@v1; mesh={l,l1,RotateLeft[l1],RotateLeft[l]}; mesh=Map[Drop[#, -1]&amp;,mesh,{2}]; pp1=Polygon/@Transpose[Map[Flatten[#,1]&amp;,mesh]]; pp2=Polygon[ Table[rcphi=N[f[[1]]]; rsphi=N[f[[2]]]; {rcphi,rsphi,f[[3]]},{i,n} ]; pp3=Polygon[</pre>	<p>Función que regresa un elemento gráfico correspondiente a un cilindro de radio <b>r</b>, requerido para dibujar los primeros elementos del PUMA.</p> <p>Genera una tabla de puntos en toda la superficie del cilindro, y se forman polígonos como superficies que conforman el cilindro.</p>

<pre> Table[rcphi=N[g[[1]]]; rsphi=N[g[[2]]];       {rcphi,rsphi,g[[3]]},{i,n} ]; pp=Append[pp1,pp2]; pp=Append[pp,pp3]; Return[pp] ]; </pre>	
<pre> Cilindro[r_,h_,o_,q_]:=Cilindro[r,h,o,q,{0,0,0}]; </pre>	<p>Dibuja un Cilindro en la dirección <math>o[1:x,2:y,3:z]</math> rotado con un Cuaternión <math>q</math> y desplazado con un vector <math>v</math></p>
<pre> Cilindro[r_,h_,o_]:=Cilindro[r,h,o,Quaternion[1,0,0,0]] </pre>	
<pre> Cilindro[r_,h_]:=Cilindro[r,h,3]; </pre>	
<pre> Sardina[R_,r_,H_,L_,o_,v_,q_]:= Module[{qq=q,i,h,n,f2,g2,rcphi,rsphi,rcphi2,       rsphi2,Lv,RR,m1,pp1,pp2,ff,gg,f,g,l1,t0,t1,       teta0,teta1,pol1,pol2,pol3,pol4,vv,v2,lv,l1},       teta0=N[ArcCos[(R-r)/L]];       teta1=2.0 <math>\pi</math>-teta0;n=10;       Switch[o,         1,Lv={0,L,0},{* x *}         2,Lv={L,0,0},{* y *}         3,Lv={L,0,0} (* z *)       ];       Switch[o,         1,ff={h,               RR Cos[i/n (tt1 - tt0)+tt0],               RR Sin[i/n (tt1 - tt0)+tt0]},{* x *}         2,ff={RR Cos[i/n (tt1 - tt0)+tt0],               h,               RR Sin[i/n (tt1 - tt0)+tt0]},{* y *}         3,ff={RR Cos[i/n (tt1 - tt0)+tt0],               RR Sin[i/n (tt1 - tt0)+tt0],               h}{* z *}       ];       t0=-teta0;t1=teta0;       gg=ff/.{RR-&gt;r,tt1-&gt;t1,tt0-&gt;t0};       ff=ff/.{RR-&gt;R,tt1-&gt;teta1,tt0-&gt;teta0}; </pre>	<p>Función que devuelve un elemento gráfico, correspondiente a un elemento del manipulador, dividiendo la superficie en puntos inicialmente, y se orienta en función de <math>Lv</math></p>

```

lv =fromQuaternion[
    RotacionQ[qq,toQuaternion[Lv]]];
vv = v;
f=vv+fromQuaternion[
    RotacionQ[qq,toQuaternion[ff/.h->H/2.0]]];
g=vv+fromQuaternion[
    RotacionQ[qq,toQuaternion[ff/.h->-H/2.0]]];
f2=lv+vv+fromQuaternion[
    RotacionQ[qq,toQuaternion[
        gg/.h->H/2.0]]];
g2=lv+vv+fromQuaternion[
    RotacionQ[qq,toQuaternion[
        gg/.h->-H/2.0]]];
v2=Table[rcphi=N[f2[[1]]];
    rsphi=N[f2[[2]]];
    rcphi2=N[g2[[1]]];
    rsphi2=N[g2[[2]]];
    {{rcphi,rsphi,f2[[3]]},
    {rcphi2,rsphi2,g2[[3]]}},{i,0,n}];
l1=v2;l12=RotateLeft/@v2;
mesh2={l1,l12,RotateLeft[l12],RotateLeft[l1]};
mesh2=Map[Drop[#, -1]&,mesh2,{1}];
pp2=Polygon/@Transpose[Map[Flatten[#,1]&,mesh2]];
v1=Table[rcphi=N[f[[1]]];
    rsphi=N[f[[2]]];
    rcphi2=N[g[[1]]];
    rsphi2=N[g[[2]]];
    {{rcphi,rsphi,f[[3]]},
    {rcphi2,rsphi2,g[[3]]}},
    {i,0,n}];
l=v1;
l1=RotateLeft/@v1;
mesh={l,l1,RotateLeft[l1],RotateLeft[l]};
mesh=Map[Drop[#, -1]&,mesh,{1}];
pp1=Polygon/@Transpose[Map[Flatten[#,1]&,mesh]];
n1=Polygon[{{f/.i->0,f2/.i->n,g2/.i->n,g/.i->0}}];
n2=Polygon[{{f/.i->n,f2/.i->0,g2/.i->0,g/.i->n}}];
m1=Table[rcphi=N[f[[1]]];
    rsphi=N[f[[2]]];
    {rcphi,rsphi,f[[3]]},{i,0,n}];
m1=Append[m1,f/.i->0];
m1=Append[m1,f2/.i->n];
m1=Append[m1,f2/.i->0];
m1=Append[m1,f/.i->n];
m2=Table[rcphi=N[f2[[1]]];
    rsphi=N[f2[[2]]];

```

<pre> {rcphi, rsphi, f2[[3]], {i, 0, n}}; pol1=Polygon[m1]; pol2=Polygon[m2]; m1=Table[rcphi=N[g[[1]]];       rsphi=N[g[[2]]];       {rcphi, rsphi, g[[3]], {i, 0, n}}; m1=Append[m1, g/.i-&gt;0]; m1=Append[m1, g2/.i-&gt;n]; m1=Append[m1, g2/.i-&gt;0]; m1=Append[m1, g/.i-&gt;n]; m2=Table[rcphi=N[g2[[1]]];       rsphi=N[g2[[2]]];       {rcphi, rsphi, g2[[3]], {i, 0, n}}; pol3=Polygon[m1]; pol4=Polygon[m2]; pp3=Polygon[Table[rcphi=N[g[[1]]];       rsphi=N[g[[2]]];       {rcphi, rsphi, g[[3]], {i, 0, n}}]; Graphics3D[{pp1, pp2, n1, n2, pol1, pol2, pol3, pol4},       Lighting-&gt;False,       ColorFunction-&gt;Hue] ]; </pre>	
<b>Simplificación de ecuaciones</b>	
<pre> Simplifica[Ec_]:=Simplify[Factor[Expand[Ec]//. RRR]//.       rrr; </pre>	Simplificación de una ecuación.
<pre> PQ[Ec1_, Ec2_]:=ProductoQ[Ec1, Ec2]; </pre>	Definición de la multiplicación de Cuaterniones
<b>Inicio de expresiones de rotación</b>	
<pre> Off[General::"spell1"] </pre>	Elimina el despliegue de advertencias para Mathematica®
<b>Generación de los sistemas de referencia</b>	
<pre> ClearAll[P, Q, R, S, T, U, p, p0, p3, q, q0, q1, r, r0, r1, s, s0, s2, t, t0,       t1, u, u0, u2, "e*"]; </pre>	Limpia las variables.
<pre> e1={1, 0, 0}; e2={0, 1, 0}; </pre>	Sistema de referencia

<pre>e3={0,0,1}; p=Quaternion[p0,0,0,p3];q=Quaternion[q0,q1,0,0]; r=Quaternion[r0,r1,0,0];s=Quaternion[s0,0,s2,0]; t=Quaternion[t0,t1,0,0];u=Quaternion[u0,0,u2,0];</pre>	<p>Definición de los Cuaterniones de rotación</p>
<pre>e11=Simplifica[Rotacion[p,toQuaternion[e1]]]; e21=Simplifica[Rotacion[p,toQuaternion[e2]]]; e31=Simplifica[Rotacion[p,toQuaternion[e3]]]; pq=p**q;</pre>	<p>Sistema de referencia del primer elemento. Cuaternión de la segunda rotación</p>
<pre>e12=Simplifica[Rotacion[pq,toQuaternion[e1]]]; e22=Simplifica[Rotacion[pq,toQuaternion[e2]]]; e32=Simplifica[Rotacion[pq,toQuaternion[e3]]]; pr=p**q**r;</pre>	<p>Sistema de referencia del tercer elemento. Cuaternión de la tercer rotación.</p>
<pre>e13=Simplifica[Rotacion[pr,toQuaternion[e1]]]; e23=Simplifica[Rotacion[pr,toQuaternion[e2]]]; e33=Simplifica[Rotacion[pr,toQuaternion[e3]]]; ps=p**q**r**s//Simplify;</pre>	<p>Sistema de referencia del tercer eslabón Cuaternión de la cuarta rotación</p>
<pre>e14=Simplifica[Rotacion[ps,toQuaternion[e1]]]; e24=Simplifica[Rotacion[ps,toQuaternion[e2]]]; e34=Simplifica[Rotacion[ps,toQuaternion[e3]]]; pt=p**q**r**s**t//Simplify;</pre>	<p>Sistema de referencia del cuarto elemento. Cuaternión de la quinta rotación.</p>
<pre>e15=Simplifica[Rotacion[pt,toQuaternion[e1]]]; e25=Simplifica[Rotacion[pt,toQuaternion[e2]]]; e35=Simplifica[Rotacion[pt,toQuaternion[e3]]]; pu=p**q**r**s**t**u//ExpandAll//Simplify;</pre>	<p>Quinto sistema de referencia. Cuaternión para la sexta rotación.</p>
<pre>e16=Simplifica[Rotacion[pu,toQuaternion[e1]]];</pre>	<p>Sexto sistema</p>



<pre>e26=Simplifica[Rotacion[pu,toQuaternion[e2]]]; e36=Simplifica[Rotacion[pu,toQuaternion[e3]]];</pre>	de referencia.
<b>Determinación de los elementos del manipulador</b>	
<p>Determinación de los eslabones</p> <pre>ClearAll[d1,d2,d3,d4,d5,d6,p0,p3,q0,q1,r0,r1,s0,s2,t0,t1, u0,u2]</pre>	Limpieza de variables
<pre>d1 = 660.4; d2 = 150; d3 = 431.8; d4 = 50; d5 = 423.07; d6 = 96.25; d7 = 50; d8 = 100;</pre>	Longitud de los eslabones.
<pre>b1 = d1 e3; b2 = d2 (-e12); b3 = d3 e22; b4 = d4 (e13); b5 = d5 e23; b6 = d6 e26;  RP = b1 + b2 + b3 + b4 + b5 + b6;  rp = b1 + b2 + b3 + b4 + b5;</pre>	Primer vector Segundo vector Tercer vector Cuarto vector Quinto elemento Elemento terminal Vector <b>RP</b> al elemento terminal Vector de los primeros 3 grados de libertad

Tabla 5.1 Código del programa

### 5.5.3 Explicitación de los ejes de rotación

Las expresiones determinadas de los diferentes ejes de rotación se explicitan de la manera siguiente:

*Primeros eje de rotación (ej1):*

$$e11 = \{-1 + 2*p0^2, 2*p0*p3, 0\}$$

$$e21 = \{-2*p0*p3, -1 + 2*p0^2, 0\}$$

$$e31 = \{0, 0, 1\}$$

*Segundos ejes de rotación (ej2):*

$$e12 = \{-1 + 2*p0^2, 2*p0*p3, 0\}$$

$$e22 = \{-2*p0*p3*(-1 + 2*q0^2), (-1 + 2*p0^2)*(-1 + 2*q0^2), 2*q0*q1\}$$

$$e32 = \{4*p0*p3*q0*q1, -2*(-1 + 2*p0^2)*q0*q1, -1 + 2*q0^2\}$$

Terceros ejes de rotación (ej3):

$$e13 = \{-1 + 2*p0^2, 2*p0*p3, 0\}$$

$$e23 = \{2*p0*p3*(-1 + 2*r0^2 + q0^2*(2 - 4*r0^2) + 4*q0*q1*r0*r1), \\ (-1 + 2*p0^2)*(1 - 2*r0^2 + q0^2*(-2 + 4*r0^2) - 4*q0*q1*r0*r1), \\ 2*q0*q1*(-1 + 2*r0^2) - 2*r0*r1 + 4*q0^2*r0*r1\}$$

$$e33 = \{4*p0*p3*(q0*q1*(-1 + 2*r0^2) - r0*r1 + 2*q0^2*r0*r1), \\ -2*(-1 + 2*p0^2)*(q0*q1*(-1 + 2*r0^2) - r0*r1 + 2*q0^2*r0*r1), \\ 1 - 2*r0^2 + q0^2*(-2 + 4*r0^2) - 4*q0*q1*r0*r1\}$$

Cuartos ejes de rotación (ej4):

$$e14 = \{1 - 2*s0^2 + p0^2*(-2 + 4*s0^2) - 8*p0*p3*(q0*q1*(-1 + 2*r0^2) - r0*r1 + 2*q0^2*r0*r1)*s0*s2, \\ 2*(p0*p3*(-1 + 2*s0^2) + 2*(q0*q1*(1 - 2*r0^2) + r0*r1 - 2*q0^2*r0*r1)*s0*s2 + \\ 4*p0^2*(q0*q1*(-1 + 2*r0^2) - r0*r1 + 2*q0^2*r0*r1)*s0*s2), 2*(-1 + 2*r0^2 + q0^2*(2 - 4*r0^2) + \\ 4*q0*q1*r0*r1)*s0*s2\}$$

$$e24 = \{2*p0*p3*(-1 + 2*r0^2 + q0^2*(2 - 4*r0^2) + 4*q0*q1*r0*r1), \\ (-1 + 2*p0^2)*(1 - 2*r0^2 + q0^2*(-2 + 4*r0^2) - 4*q0*q1*r0*r1), \\ 2*q0*q1*(-1 + 2*r0^2) - 2*r0*r1 + 4*q0^2*r0*r1\}$$

$$e34 = \{2*(2*p0*p3*(q0*q1*(-1 + 2*r0^2) - r0*r1 + 2*q0^2*r0*r1)*(-1 + 2*s0^2) - s0*s2 + 2*p0^2*s0*s2), \\ 2*(-((-1 + 2*p0^2)*q0*q1*(-1 + 2*r0^2)*(-1 + 2*s0^2)) + (-1 + 2*p0^2)*r0*r1*(-1 + 2*s0^2) - 2*(-1 \\ + 2*p0^2)*q0^2*r0*r1*(-1 + 2*s0^2) + 2*p0*p3*s0*s2), (1 - 2*r0^2 + q0^2*(-2 + 4*r0^2) - \\ 4*q0*q1*r0*r1)*(-1 + 2*s0^2)\}$$

Quintos ejes de rotación (ej5):

$$e15 = \{1 - 2*s0^2 + p0^2*(-2 + 4*s0^2) - 8*p0*p3*(q0*q1*(-1 + 2*r0^2) - r0*r1 + 2*q0^2*r0*r1)*s0*s2, \\ 2*(p0*p3*(-1 + 2*s0^2) + 2*(q0*q1*(1 - 2*r0^2) + r0*r1 - 2*q0^2*r0*r1)*s0*s2 + \\ 4*p0^2*(q0*q1*(-1 + 2*r0^2) - r0*r1 + 2*q0^2*r0*r1)*s0*s2), 2*(-1 + 2*r0^2 + q0^2*(2 - 4*r0^2) + \\ 4*q0*q1*r0*r1)*s0*s2\}$$

$$e25 = \{2*(-2*s0*s2*t0*t1 + 4*p0^2*s0*s2*t0*t1 + p0*p3*(1 - 2*t0^2 + r0^2*(-2 + 4*t0^2) + 4*r0*r1*(1 - \\ 2*s0^2)*t0*t1 + 4*q0*q1*(r0*r1*(-1 + 2*t0^2) + (1 - 2*s0^2)*t0*t1 + 2*r0^2*(-1 + 2*s0^2)*t0*t1) + \\ q0^2*(-2 + 4*t0^2 + r0^2*(4 - 8*t0^2) + 8*r0*r1*(-1 + 2*s0^2)*t0*t1))), 1 - 2*r0^2 - 2*t0^2 + 4*r0^2*t0^2 \\ + 4*r0*r1*t0*t1 - 8*r0*r1*s0^2*t0*t1 + 8*p0*p3*s0*s2*t0*t1 + 4*q0*q1*(r0*r1*(-1 + 2*t0^2) + \\ (1 - 2*s0^2)*t0*t1 + 2*r0^2*(-1 + 2*s0^2)*t0*t1) + q0^2*(-2 + 4*t0^2 + r0^2*(4 - 8*t0^2) + 8*r0*r1*(-1 \\ + 2*s0^2)*t0*t1) - 2*p0^2*(1 - 2*t0^2 + r0^2*(-2 + 4*t0^2) + 4*r0*r1*(1 - 2*s0^2)*t0*t1 + \\ 4*q0*q1*(r0*r1*(-1 + 2*t0^2) + (1 - 2*s0^2)*t0*t1 + 2*r0^2*(-1 + 2*s0^2)*t0*t1) + q0^2*(-2 + 4*t0^2 \\ + r0^2*(4 - 8*t0^2) + 8*r0*r1*(-1 + 2*s0^2)*t0*t1)), -2*(r0*r1*(-1 + 2*t0^2) + (1 - 2*s0^2)*t0*t1 + \\ 2*r0^2*(-1 + 2*s0^2)*t0*t1 - q0*q1*(1 - 2*t0^2 + r0^2*(-2 + 4*t0^2) + 4*r0*r1*(1 - 2*s0^2)*t0*t1) + \\ q0^2*(2*r0*r1*(1 - 2*t0^2) + 4*r0^2*(1 - 2*s0^2)*t0*t1 + 2*(-1 + 2*s0^2)*t0*t1))\}$$

$$e35 = \{-2*(2*p0^2*s0*s2*(1 - 2*t0^2) + s0*s2*(-1 + 2*t0^2) + 2*p0*p3*(r0*r1*(-1 + 2*s0^2)*(-1 + 2*t0^2) - \\ t0*t1 + 2*r0^2*t0*t1 + q0^2*(-2*r0*r1*(-1 + 2*s0^2)*(-1 + 2*t0^2) + 2*t0*t1 - 4*r0^2*t0*t1) + \\ q0*q1*((-1 + 2*s0^2)*(-1 + 2*t0^2) - 2*r0^2*(-1 + 2*s0^2)*(-1 + 2*t0^2) + 4*r0*r1*t0*t1))), 2*(- \\ 2*p0*p3*s0*s2 + 4*p0*p3*s0*s2*t0^2 + (-1 + 2*p0^2)*r0*r1*(-1 + 2*s0^2)*(-1 + 2*t0^2) + t0*t1 - \\ 2*p0^2*t0*t1 + 2*(-1 + 2*p0^2)*r0^2*t0*t1 - 2*(-1 + 2*p0^2)*q0^2*(r0*r1*(-1 + 2*s0^2)*(-1 + 2*t0^2)$$

$$- t0*t1 + 2*r0^2*t0*t1) - (-1 + 2*p0^2)*q0*q1*(-((-1 + 2*s0^2)*(-1 + 2*t0^2)) + 2*r0^2*(-1 + 2*s0^2)*(-1 + 2*t0^2) - 4*r0*r1*t0*t1)), (-1 + 2*s0^2)*(-1 + 2*t0^2) - 2*r0^2*(-1 + 2*s0^2)*(-1 + 2*t0^2) + 4*r0*r1*t0*t1 - 4*q0*q1*(r0*r1*(-1 + 2*s0^2)*(-1 + 2*t0^2) - t0*t1 + 2*r0^2*t0*t1) + 2*q0^2*(-((-1 + 2*s0^2)*(-1 + 2*t0^2)) + 2*r0^2*(-1 + 2*s0^2)*(-1 + 2*t0^2) - 4*r0*r1*t0*t1))$$

Sextos ejes de rotación (ej6):

$$e16 = \{-1 + 2*u0^2 + s0^2(2 - 4*u0^2) + 4*s0*s2*(-1 + 2*t0^2)*u0*u2 + 2*p0^2(1 - 2*u0^2 + s0^2(-2 + 4*u0^2) + 4*s0*s2*(1 - 2*t0^2)*u0*u2) - 8*p0*p3*(t0*t1*u0*u2 - 2*r0^2*t0*t1*u0*u2 + r0*r1*(s0*s2*(1 - 2*u0^2) + 2*s0^2*(1 - 2*t0^2)*u0*u2 + (-1 + 2*t0^2)*u0*u2) + q0*q1*(-(1 + 2*r0^2)*s0*s2*(-1 + 2*u0^2) + 2*(-1 + 2*r0^2)*s0^2*(-1 + 2*t0^2)*u0*u2 + (-1 + 2*t0^2 + r0^2(2 - 4*t0^2) - 4*r0*r1*t0*t1)*u0*u2) + 2*q0^2*(-(t0*t1*u0*u2) + 2*r0^2*t0*t1*u0*u2 + r0*r1*(s0*s2*(-1 + 2*u0^2) + (1 - 2*t0^2)*u0*u2 + 2*s0^2*(-1 + 2*t0^2)*u0*u2)), 2*(p0*p3*(1 - 2*u0^2 + s0^2(-2 + 4*u0^2) + 4*s0*s2*(1 - 2*t0^2)*u0*u2) - 2*(t0*t1*u0*u2 - 2*r0^2*t0*t1*u0*u2 + r0*r1*(s0*s2*(1 - 2*u0^2) + 2*s0^2*(1 - 2*t0^2)*u0*u2 + (-1 + 2*t0^2)*u0*u2) + q0*q1*(-(1 + 2*r0^2)*s0*s2*(-1 + 2*u0^2) + 2*(-1 + 2*r0^2)*s0^2*(-1 + 2*t0^2)*u0*u2 + (-1 + 2*t0^2 + r0^2(2 - 4*t0^2) - 4*r0*r1*t0*t1)*u0*u2) + 2*q0^2*(-(t0*t1*u0*u2) + 2*r0^2*t0*t1*u0*u2 + r0*r1*(s0*s2*(-1 + 2*u0^2) + (1 - 2*t0^2)*u0*u2 + 2*s0^2*(-1 + 2*t0^2)*u0*u2))) + 4*p0^2*(t0*t1*u0*u2 - 2*r0^2*t0*t1*u0*u2 + r0*r1*(s0*s2*(1 - 2*u0^2) + 2*s0^2*(1 - 2*t0^2)*u0*u2 + (-1 + 2*t0^2)*u0*u2) + q0*q1*(-(1 + 2*r0^2)*s0*s2*(-1 + 2*u0^2) + 2*(-1 + 2*r0^2)*s0^2*(-1 + 2*t0^2)*u0*u2 + (-1 + 2*t0^2 + r0^2(2 - 4*t0^2) - 4*r0*r1*t0*t1)*u0*u2) + 2*q0^2*(-(t0*t1*u0*u2) + 2*r0^2*t0*t1*u0*u2 + r0*r1*(s0*s2*(-1 + 2*u0^2) + (1 - 2*t0^2)*u0*u2 + 2*s0^2*(-1 + 2*t0^2)*u0*u2))), 2*(-((1 - 2*r0^2 + q0^2(-2 + 4*r0^2) - 4*q0*q1*r0*r1)*s0*s2*(-1 + 2*u0^2)) - 2*(1 - 2*r0^2 + q0^2(-2 + 4*r0^2) - 4*q0*q1*r0*r1)*s0^2*(-1 + 2*t0^2)*u0*u2 + (-1 + 2*t0^2 + r0^2(2 - 4*t0^2) - 4*r0*r1*t0*t1 + 4*q0*q1*(r0*r1*(1 - 2*t0^2) - t0*t1 + 2*r0^2*t0*t1) + q0^2(2 - 4*t0^2 + r0^2(-4 + 8*t0^2) + 8*r0*r1*t0*t1))*u0*u2)}$$

$$e26 = \{2*(-2*s0*s2*t0*t1 + 4*p0^2*s0*s2*t0*t1 + p0*p3*(1 - 2*t0^2 + r0^2(-2 + 4*t0^2) + 4*r0*r1*(1 - 2*s0^2)*t0*t1 + 4*q0*q1*(r0*r1*(-1 + 2*t0^2) + (1 - 2*s0^2)*t0*t1 + 2*r0^2*(-1 + 2*s0^2)*t0*t1) + q0^2(-2 + 4*t0^2 + r0^2(4 - 8*t0^2) + 8*r0*r1*(-1 + 2*s0^2)*t0*t1)), 1 - 2*r0^2 - 2*t0^2 + 4*r0^2*t0^2 + 4*r0*r1*t0*t1 - 8*r0*r1*s0^2*t0*t1 + 8*p0*p3*s0*s2*t0*t1 + 4*q0*q1*(r0*r1*(-1 + 2*t0^2) + (1 - 2*s0^2)*t0*t1 + 2*r0^2*(-1 + 2*s0^2)*t0*t1) + q0^2(-2 + 4*t0^2 + r0^2(4 - 8*t0^2) + 8*r0*r1*(-1 + 2*s0^2)*t0*t1) - 2*p0^2(1 - 2*t0^2 + r0^2(-2 + 4*t0^2) + 4*r0*r1*(1 - 2*s0^2)*t0*t1 + 4*q0*q1*(r0*r1*(-1 + 2*t0^2) + (1 - 2*s0^2)*t0*t1 + 2*r0^2*(-1 + 2*s0^2)*t0*t1) + q0^2(-2 + 4*t0^2 + r0^2(4 - 8*t0^2) + 8*r0*r1*(-1 + 2*s0^2)*t0*t1)), -2*(r0*r1*(-1 + 2*t0^2) + (1 - 2*s0^2)*t0*t1 + 2*r0^2*(-1 + 2*s0^2)*t0*t1 - q0*q1*(1 - 2*t0^2 + r0^2(-2 + 4*t0^2) + 4*r0*r1*(1 - 2*s0^2)*t0*t1) + q0^2(2*r0*r1*(1 - 2*t0^2) + 4*r0^2*(1 - 2*s0^2)*t0*t1 + 2*(-1 + 2*s0^2)*t0*t1))\}$$

$$e36 = \{-2*(s0*s2*(-1 + 2*t0^2)*(-1 + 2*u0^2) - u0*u2 + 2*s0^2*u0*u2 + p0^2(-2*s0*s2*(-1 + 2*t0^2)*(-1 + 2*u0^2) + 2*u0*u2 - 4*s0^2*u0*u2) + 2*p0*p3*(t0*t1*(1 - 2*u0^2) + 2*r0^2*t0*t1*(-1 + 2*u0^2) + r0*r1*(-((-1 + 2*t0^2)*(-1 + 2*u0^2)) + 2*s0^2*(-1 + 2*t0^2)*(-1 + 2*u0^2) - 4*s0*s2*u0*u2) - q0*q1*(4*r0*r1*t0*t1*(1 - 2*u0^2) + (-1 + 2*t0^2)*(-1 + 2*u0^2) - 2*s0^2*(-1 + 2*t0^2)*(-1 + 2*u0^2) + 4*s0*s2*u0*u2 + 2*r0^2*(-((-1 + 2*t0^2)*(-1 + 2*u0^2)) + 2*s0^2*(-1 + 2*t0^2)*(-1 + 2*u0^2) + 4*s0*s2*u0*u2) - 2*(t0*t1*u0*u2 - 2*r0^2*t0*t1*u0*u2 + r0*r1*(s0*s2*(1 - 2*u0^2) + 2*s0^2*(1 - 2*t0^2)*u0*u2 + (-1 + 2*t0^2)*u0*u2) + q0*q1*(-(1 + 2*r0^2)*s0*s2*(-1 + 2*u0^2) + 2*(-1 + 2*r0^2)*s0^2*(-1 + 2*t0^2)*u0*u2 + (-1 + 2*t0^2 + r0^2(2 - 4*t0^2) - 4*r0*r1*t0*t1)*u0*u2) + 2*q0^2*(-(t0*t1*u0*u2) + 2*r0^2*t0*t1*u0*u2 + r0*r1*(s0*s2*(-1 + 2*u0^2) + (1 - 2*t0^2)*u0*u2 + 2*s0^2*(-1 + 2*t0^2)*u0*u2))), 2*(-((1 - 2*r0^2 + q0^2(-2 + 4*r0^2) - 4*q0*q1*r0*r1)*s0*s2*(-1 + 2*u0^2)) - 2*(1 - 2*r0^2 + q0^2(-2 + 4*r0^2) - 4*q0*q1*r0*r1)*s0^2*(-1 + 2*t0^2)*u0*u2 + (-1 + 2*t0^2 + r0^2(2 - 4*t0^2) - 4*r0*r1*t0*t1 + 4*q0*q1*(r0*r1*(1 - 2*t0^2) - t0*t1 + 2*r0^2*t0*t1) + q0^2(2 - 4*t0^2 + r0^2(-4 + 8*t0^2) + 8*r0*r1*t0*t1))*u0*u2)}$$

$$\begin{aligned}
& 2^*u0^2) - 4^*s0*s2^*u0^*u2)) + q0^{2^*}(4^*r0^{2^*}t0^*t1^*(1 - 2^*u0^2) + 2^*t0^*t1^*(-1 + 2^*u0^2) + 2^*r0^*r1^*((- \\
& 1 + 2^*t0^2)^*(-1 + 2^*u0^2) - 2^*s0^{2^*}(-1 + 2^*t0^2)^*(-1 + 2^*u0^2) + 4^*s0^*s2^*u0^*u2))), \\
& 2^*(2^*p0^*p3^*s0^*s2 - 4^*p0^*p3^*s0^*s2^*t0^2 - t0^*t1 + 2^*p0^{2^*}t0^*t1 - 4^*p0^*p3^*s0^*s2^*u0^2 + \\
& 8^*p0^*p3^*s0^*s2^*t0^2^*u0^2 + 2^*t0^*t1^*u0^2 - 4^*p0^{2^*}t0^*t1^*u0^2 + 2^*(-1 + 2^*p0^2)^*r0^{2^*}t0^*t1^*(-1 + \\
& 2^*u0^2) - 2^*p0^*p3^*u0^*u2 + 4^*p0^*p3^*s0^{2^*}u0^*u2 + (-1 + 2^*p0^2)^*r0^*r1^*((-1 + 2^*t0^2)^*(-1 + \\
& 2^*u0^2)) + 2^*s0^{2^*}(-1 + 2^*t0^2)^*(-1 + 2^*u0^2) - 4^*s0^*s2^*u0^*u2) - (-1 + \\
& 2^*p0^2)^*q0^*q1^*(4^*r0^*r1^*t0^*t1^*(1 - 2^*u0^2) + (-1 + 2^*t0^2)^*(-1 + 2^*u0^2) - 2^*s0^{2^*}(-1 + 2^*t0^2)^*(-1 \\
& + 2^*u0^2) + 4^*s0^*s2^*u0^*u2 + 2^*r0^{2^*}(-(-1 + 2^*t0^2)^*(-1 + 2^*u0^2)) + 2^*s0^{2^*}(-1 + 2^*t0^2)^*(-1 + \\
& 2^*u0^2) - 4^*s0^*s2^*u0^*u2) - 2^*(-1 + 2^*p0^2)^*q0^{2^*}(t0^*t1^*(1 - 2^*u0^2) + 2^*r0^{2^*}t0^*t1^*(-1 + 2^*u0^2) \\
& + r0^*r1^*((-1 + 2^*t0^2)^*(-1 + 2^*u0^2)) + 2^*s0^{2^*}(-1 + 2^*t0^2)^*(-1 + 2^*u0^2) - 4^*s0^*s2^*u0^*u2))), \\
& -1 + 2^*s0^2 + 2^*t0^2 - 4^*s0^{2^*}t0^2 + 2^*u0^2 - 4^*s0^{2^*}u0^2 - 4^*t0^2^*u0^2 + 8^*s0^{2^*}t0^{2^*}u0^2 + \\
& 4^*r0^*r1^*t0^*t1^*(-1 + 2^*u0^2) - 4^*s0^*s2^*u0^*u2 + r0^{2^*}(2^*(-1 + 2^*t0^2)^*(-1 + 2^*u0^2) - 4^*s0^{2^*}(-1 + \\
& 2^*t0^2)^*(-1 + 2^*u0^2) + 8^*s0^*s2^*u0^*u2) + 2^*q0^{2^*}(4^*r0^*r1^*t0^*t1^*(1 - 2^*u0^2) + (-1 + 2^*t0^2)^*(-1 \\
& + 2^*u0^2) - 2^*s0^{2^*}(-1 + 2^*t0^2)^*(-1 + 2^*u0^2) + 4^*s0^*s2^*u0^*u2 + 2^*r0^{2^*}(-(-1 + 2^*t0^2)^*(-1 + \\
& 2^*u0^2)) + 2^*s0^{2^*}(-1 + 2^*t0^2)^*(-1 + 2^*u0^2) - 4^*s0^*s2^*u0^*u2) - 4^*q0^*q1^*(t0^*t1^*(1 - 2^*u0^2) + \\
& 2^*r0^{2^*}t0^*t1^*(-1 + 2^*u0^2) + r0^*r1^*((-1 + 2^*t0^2)^*(-1 + 2^*u0^2)) + 2^*s0^{2^*}(-1 + 2^*t0^2)^*(-1 + \\
& 2^*u0^2) - 4^*s0^*s2^*u0^*u2))}
\end{aligned}$$

### 5.5.4 Explicitación de los vectores de posición

Se puede observar la simplicidad de las ecuaciones generadas. De acuerdo con lo anterior, las ecuaciones del vector de posición “**rp**” (Ver figura 5.3) de los primeros 3 grados de libertad son:

$$\mathbf{rp} = \{150^*(1 - 2^*p0^2) + 50^*(-1 + 2^*p0^2) - 863.6^*p0^*p3^*(-1 + 2^*q0^2) + 846.14^*p0^*p3^*(-1 + 2^*r0^2 + q0^{2^*}(2 - 4^*r0^2) + 4^*q0^*q1^*r0^*r1), -200^*p0^*p3 + 431.8^*(-1 + 2^*p0^2)^*(-1 + 2^*q0^2) + 423.07^*(-1 + 2^*p0^2)^*(1 - 2^*r0^2 + q0^{2^*}(-2 + 4^*r0^2) - 4^*q0^*q1^*r0^*r1), 660.4 + 863.6^*q0^*q1 + 423.07^*(2^*q0^*q1^*(-1 + 2^*r0^2) - 2^*r0^*r1 + 4^*q0^{2^*}r0^*r1)\}$$

Por otro lado, para el caso del vector **RP** se tiene que:

$$\begin{aligned}
\mathbf{RP} = & \{150^*(1 - 2^*p0^2) + 50^*(-1 + 2^*p0^2) - 863.6^*p0^*p3^*(-1 + 2^*q0^2) + 846.14^*p0^*p3^*(-1 + 2^*r0^2 + \\
& q0^{2^*}(2 - 4^*r0^2) + 4^*q0^*q1^*r0^*r1) + 192.5^*(-2^*s0^*s2^*t0^*t1 + 4^*p0^{2^*}s0^*s2^*t0^*t1 + p0^*p3^*(1 - \\
& 2^*t0^2 + r0^{2^*}(-2 + 4^*t0^2) + 4^*r0^*r1^*(1 - 2^*s0^2)^*t0^*t1 + 4^*q0^*q1^*(r0^*r1^*(-1 + 2^*t0^2) + (1 - \\
& 2^*s0^2)^*t0^*t1 + 2^*r0^{2^*}(-1 + 2^*s0^2)^*t0^*t1) + q0^{2^*}(-2 + 4^*t0^2 + r0^{2^*}(4 - 8^*t0^2) + 8^*r0^*r1^*(-1 + \\
& 2^*s0^2)^*t0^*t1))), \\
& -200^*p0^*p3 + 431.8^*(-1 + 2^*p0^2)^*(-1 + 2^*q0^2) + 423.07^*(-1 + 2^*p0^2)^*(1 - 2^*r0^2 + q0^{2^*}(-2 + \\
& 4^*r0^2) - 4^*q0^*q1^*r0^*r1) + 96.25^*(1 - 2^*r0^2 - 2^*t0^2 + 4^*r0^{2^*}t0^2 + 4^*r0^*r1^*t0^*t1 - \\
& 8^*r0^*r1^*s0^{2^*}t0^*t1 + 8^*p0^*p3^*s0^*s2^*t0^*t1 + 4^*q0^*q1^*(r0^*r1^*(-1 + 2^*t0^2) + (1 - 2^*s0^2)^*t0^*t1 + \\
& 2^*r0^{2^*}(-1 + 2^*s0^2)^*t0^*t1) + q0^{2^*}(-2 + 4^*t0^2 + r0^{2^*}(4 - 8^*t0^2) + 8^*r0^*r1^*(-1 + 2^*s0^2)^*t0^*t1) - \\
& 2^*p0^{2^*}(1 - 2^*t0^2 + r0^{2^*}(-2 + 4^*t0^2) + 4^*r0^*r1^*(1 - 2^*s0^2)^*t0^*t1 + 4^*q0^*q1^*(r0^*r1^*(-1 + 2^*t0^2) +
\end{aligned}$$

$$(1 - 2*s0^2)*t0*t1 + 2*r0^2*(-1 + 2*s0^2)*t0*t1) + q0^2*(-2 + 4*t0^2 + r0^2*(4 - 8*t0^2) + 8*r0*r1*(-1 + 2*s0^2)*t0*t1)),$$

$$660.4 + 863.6*q0*q1 + 423.07*(2*q0*q1*(-1 + 2*r0^2) - 2*r0*r1 + 4*q0^2*r0*r1) - 192.5*(r0*r1*(-1 + 2*t0^2) + (1 - 2*s0^2)*t0*t1 + 2*r0^2*(-1 + 2*s0^2)*t0*t1 - q0*q1*(1 - 2*t0^2 + r0^2*(-2 + 4*t0^2) + 4*r0*r1*(1 - 2*s0^2)*t0*t1) + q0^2*(2*r0*r1*(1 - 2*t0^2) + 4*r0^2*(1 - 2*s0^2)*t0*t1 + 2*(-1 + 2*s0^2)*t0*t1))}$$

### 5.5.5 Salidas gráficas

La simulación de la cinemática directa utiliza la secuencia del movimiento que se realiza de la siguiente forma: cada elemento se mueve desde un ángulo mínimo a un ángulo máximo, cada incremento del ángulo es constante. La tabla 5.2 muestra el código de las salidas gráficas.

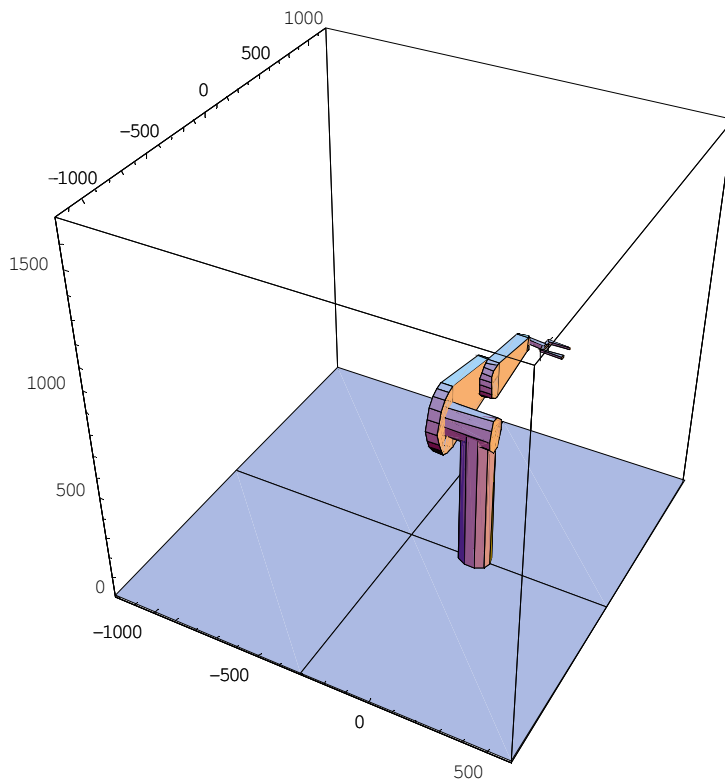
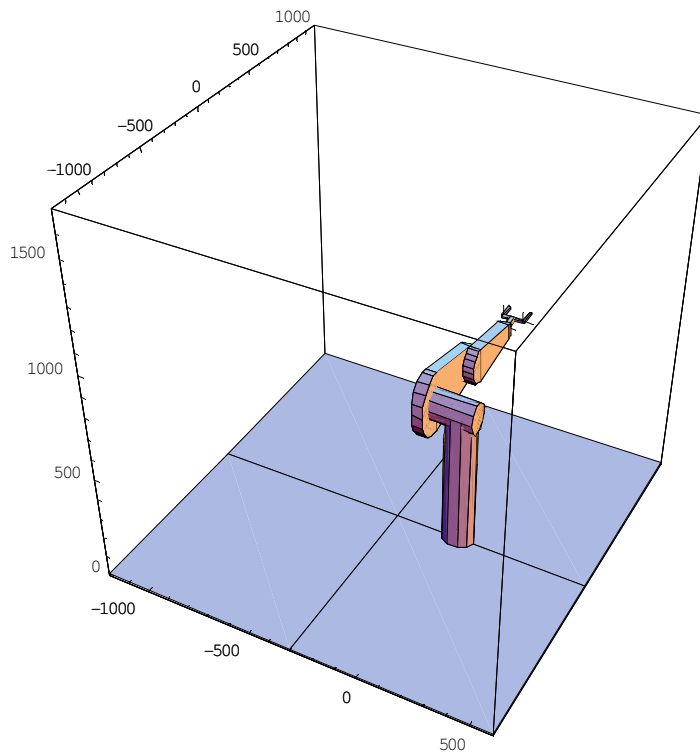
Simulación de la cinemática directa	
ClearAll[p0,p3,q0,q1,r0,r1,s0,s2,t0,t1,u0,u2];	
rad=75;	Radio del primer elemento dibujado como cilindro
anch=50;	Ancho de los elementos 2 y 3
NP=4;	NP indica la cantidad de movimientos entre el ángulo mínimo y el ángulo máximo que realiza cada elemento
$\phi_1 \text{ min} = 0; \phi_1 \text{ max} = 90;$ $\phi_2 \text{ min} = 0; \phi_2 \text{ max} = 90;$ $\phi_3 \text{ min} = 0; \phi_3 \text{ max} = 90;$ $\phi_4 \text{ min} = 0; \phi_4 \text{ max} = 90;$ $\phi_5 \text{ min} = 0; \phi_5 \text{ max} = 90;$ $\phi_6 \text{ min} = 0; \phi_6 \text{ max} = 90;$	Ángulos mínimos y ángulos máximos para cada eslabón
$\phi_1 = \phi_1 \text{ min};$ $\phi_2 = \phi_2 \text{ min};$	Inicialización del ángulo con el

$\phi_3 = \phi_{3 \text{ min}};$ $\phi_4 = \phi_{4 \text{ min}};$ $\phi_5 = \phi_{5 \text{ min}};$ $\phi_6 = \phi_{6 \text{ min}};$	<p>ángulo mínimo correspondiente.</p>
$l_1 = 20; l_2 = 50;$	
Superficie = SurfaceGraphics[{{0,0,0},{0,0,0},{0,0,0}}, MeshRange→{{-1200,600}, {-1200,1200}}, BoxRatios→{1,1,1}, Axes→True];	Superficie base donde se grafica el manipulador
For[j=6, j>0, j--, For[i=0, i≤NP, i++, Switch[j,	Ciclo para el movimiento de los elementos del manipulador, <b>j</b> indica el eslabón a mover en cada movimiento, <b>i</b> , determina el ángulo a mover
1, $\phi_1 = \phi_{1 \text{ min}} + \frac{\phi_{1 \text{ max}} - \phi_{1 \text{ min}}}{NP} i,$ 2, $\phi_2 = \phi_{2 \text{ min}} + \frac{\phi_{2 \text{ max}} - \phi_{2 \text{ min}}}{NP} i,$ 3, $\phi_3 = \phi_{3 \text{ min}} + \frac{\phi_{3 \text{ max}} - \phi_{3 \text{ min}}}{NP} i,$ 4, $\phi_4 = \phi_{4 \text{ min}} + \frac{\phi_{4 \text{ max}} - \phi_{4 \text{ min}}}{NP} i,$ 5, $\phi_5 = \phi_{5 \text{ min}} + \frac{\phi_{5 \text{ max}} - \phi_{5 \text{ min}}}{NP} i,$ 6, $\phi_6 = \phi_{6 \text{ min}} + \frac{\phi_{6 \text{ max}} - \phi_{6 \text{ min}}}{NP} i];$	Si <b>j = 1</b> , entonces se determina el ángulo del eslabón <b>1</b> , el cual es proporcional a <b>i</b>
ClearAll[p0,p3,q0,q1,r0,r1,s0,s2,t0,t1,u0,u2];	
$p_0 = \text{Cos}\left[\frac{\phi_1}{2}\right]; p_3 = \text{Sin}\left[\frac{\phi_1}{2}\right];$	Inicialización del cuaternión p
$q_0 = \text{Cos}\left[\frac{\phi_2}{2}\right]; q_1 = \text{Sin}\left[\frac{\phi_2}{2}\right];$	Inicialización del cuaternión q
$r_0 = \text{Cos}\left[\frac{\phi_3}{2}\right]; r_1 = \text{Sin}\left[\frac{\phi_3}{2}\right];$	Inicialización del cuaternión r
$s_0 = \text{Cos}\left[\frac{\phi_4}{2}\right]; s_2 = \text{Sin}\left[\frac{\phi_4}{2}\right];$	Inicialización del cuaternión s
$t_0 = \text{Cos}\left[\frac{\phi_5}{2}\right]; t_1 = \text{Sin}\left[\frac{\phi_5}{2}\right];$	Inicialización del cuaternión t

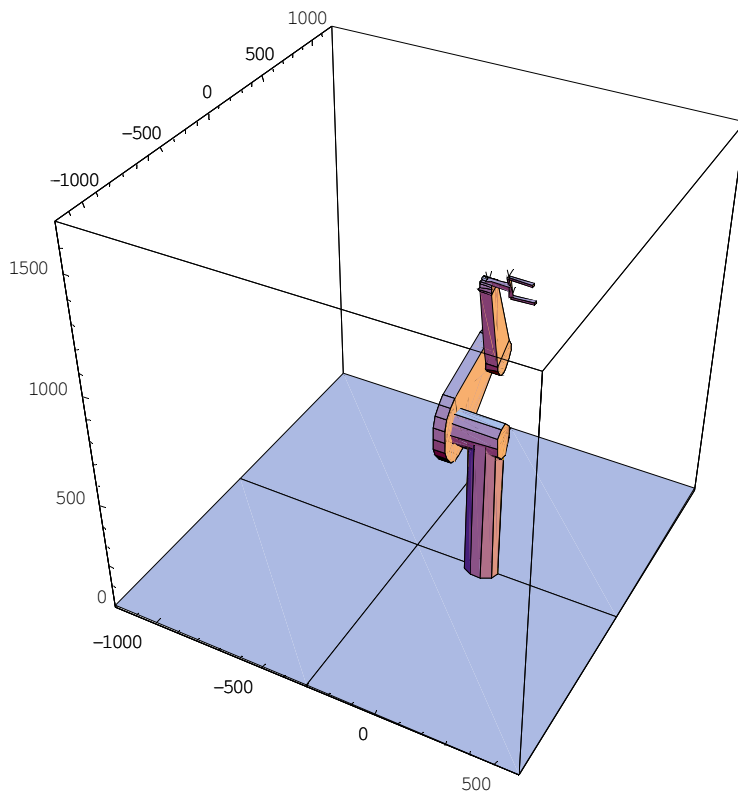
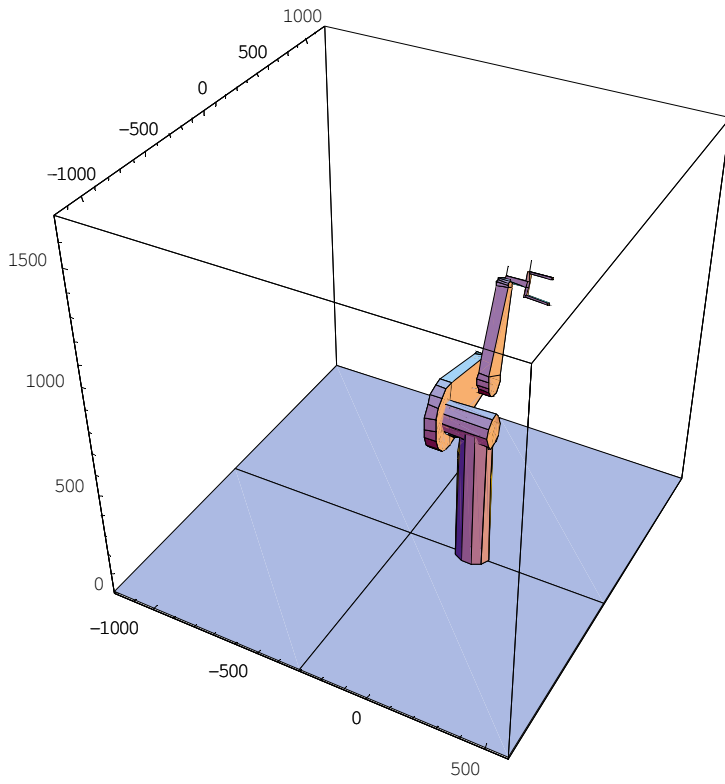
$u0 = \text{Cos}\left[\frac{\phi_6^c}{2}\right]; u2 = \text{Sin}\left[\frac{\phi_6^c}{2}\right];$	Inicialización del cuaternión $u$
Punto1={0,0,0}; Punto2=N[b1]; Punto3=N[Punto2+b2]; Punto4=N[Punto3+b3]; Punto5=N[Punto4+b4]; Punto6=N[Punto5+b5];	Definición de los puntos finales de cada vector $b$ .
b7=d7 e16; b8=d8 e26;	Vectores que definen a los elementos que determinan el órgano terminal.
Punto7=N[Punto6+b6]; Punto8=N[Punto7+b7]; Punto9=N[Punto7-b7]; Punto10=N[Punto8+b8]; Punto11=N[Punto9+b8];	Elementos del organo terminal.
Show[Superficie,  Cilindro[rad,L1], Cilindro[rad,rad+L2,-1,p,{rad,0,L1}]], Sardina[2.0 rad, 1.1 rad, anch,L3, 1, Punto3, pq], Sardina[1.1 rad, 0.5rad, anch,L5-0.5 rad, 1, Punto5, pr], Elemento[Punto6,Punto7,l4,1,e16,e26,e36], Elemento[Punto8,Punto9,l4/2,0,e16,e26,e36], Elemento[Punto8,Punto10,l4/2,1,e16,e26,e36], Elemento[Punto9,Punto11,l4/2,1,e16,e26,e36] .PlotRange→{{-1200,600},{-1200,1200},{-5,1700}}]]]	Muestra gráficamente el manipulador. Primer elemento Segundo elemento Tercer elemento Cuarto elemento Quinto elemento Partes del órgano terminal.  Rango de observación.

Tabla 5.2 Código del programa para las salidas gráficas

Finalmente, la figura 5.6 muestra una serie de salidas gráficas asociadas con el problema cinemático directo relacionado con el robot PUMA 560.







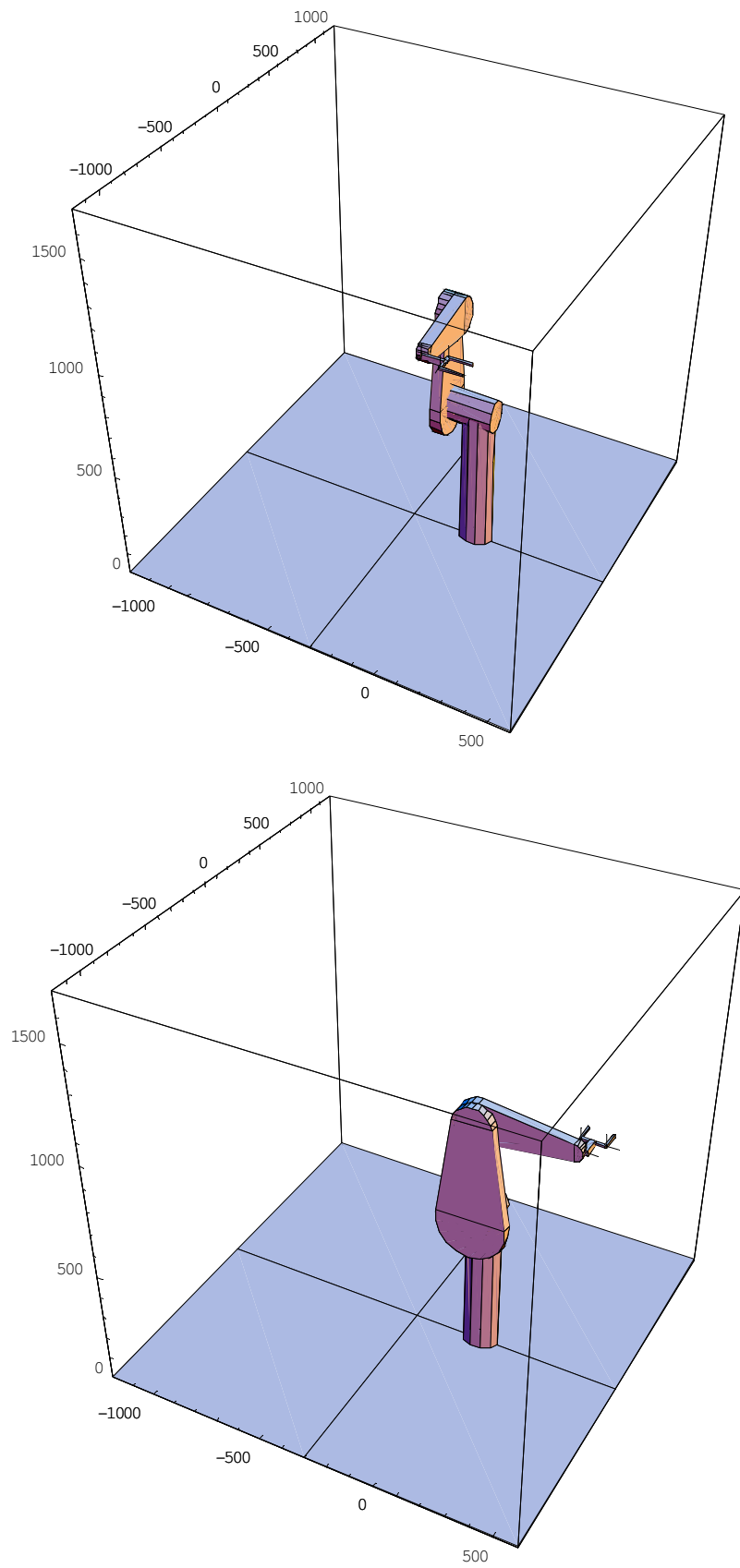


Figura 5.6 Salidas gráficas del problema cinemático directo.

## 5.6 Cinemática inversa

La complejidad de la cinemática inversa para los manipuladores existentes varía en función de su geometría:

- Para manipuladores seriales es compleja, no lineal y con múltiples soluciones y para el caso de un manipulador serial de 6 grados de libertad tiene 16 soluciones.
- Para manipuladores paralelos, se resuelve para cada cadena cerrada y es más simple que la cinemática directa.

El problema cinemático inverso para el manipulador PUMA representa el principal problema a resolver en esta tesis y no existen muchas especificaciones para su solución. Existen diversas metodologías de resolución del problema cinemático inverso, entre las que destacan:

- Métodos geométricos
- Matriz de transformación homogénea
- Desacople cinemático

En este trabajo se utiliza el desacople cinemático, el cual consiste en encontrar primero la solución para los primeros 3 grados de libertad sin considerar los 3 últimos, conocida también como posición, lo cual simplifica el modelo a resolver. La solución de los últimos 3 grados de libertad se encuentra entonces para cada posición encontrada. Un análisis de las singularidades para este tipo de manipulador se encuentra en [14] .

### 5.6.1 Algunas consideraciones sobre la trayectoria y el desarrollo del problema cinemático inverso

En esta sección se desarrolla el modelo que permitirá formular el problema cinemático inverso. Para ello, se describe la trayectoria que el órgano terminal del robot ejecutará que en este caso es una hélice. Se discutirá el procedimiento de desacoplamiento del modelo del robot.

### 5.6.2 Trayectoria de operación

En esta sección se describirá los elementos primitivos que integran la trayectoria que será modelada en este trabajo de tesis. Considere la siguiente función:

$$C : \mathcal{R} \rightarrow \mathcal{R}^3$$

Definida por:

$$f(v) = \{(h + r \cos(v)), (k + 40v), (l + r \sin(v))\} \quad (5.19)$$

Una curva paramétrica, donde  $(h, k, l)$  son las coordenadas del centro de la hélice,  $r$  es el radio y “ $v$ ” es el parámetro. La figura geométrica relacionada con la curva definida anteriormente se muestra en la figura 5.7. Por otro lado, la figura 5.8 muestra un triedro móvil asociado con la hélice.

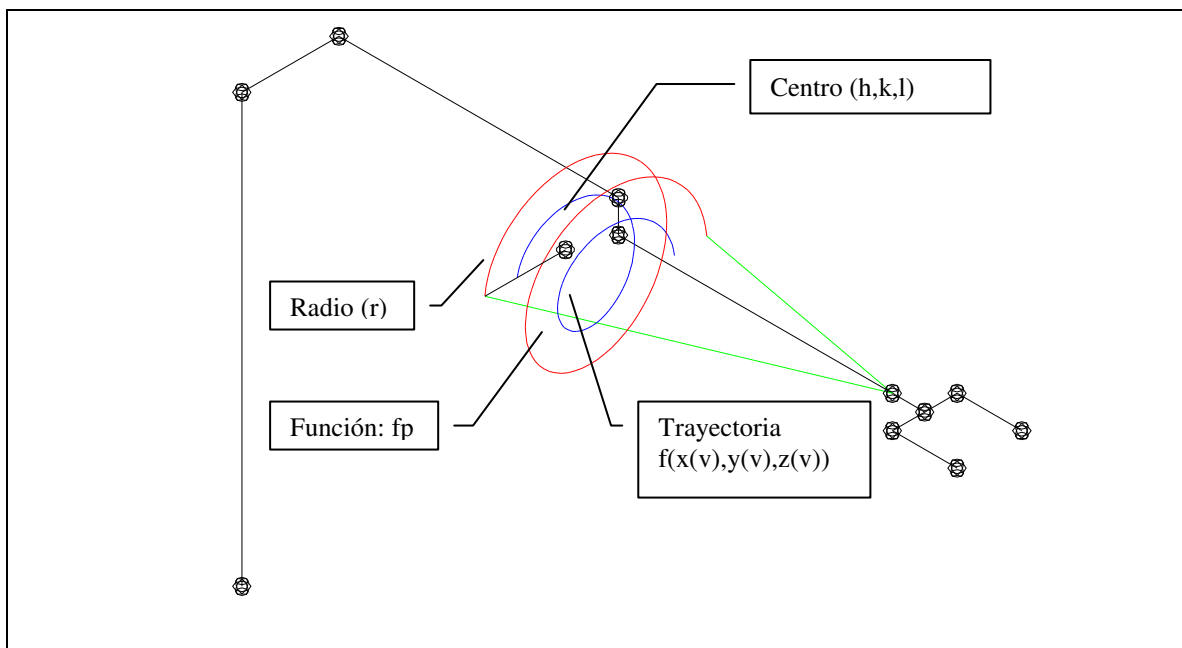


Figura 5.7 Curva paramétrica con el robot

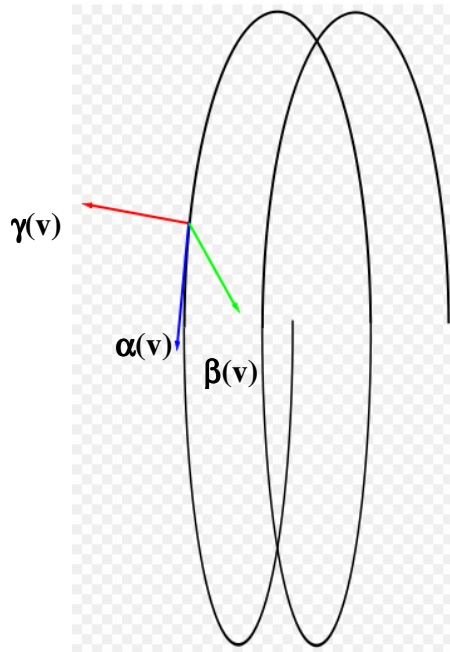


Figura 5.8 Curva paramétrica y triedro móvil

El vector  $\alpha(v)$  mostrado en la figura 5.8 es llamado “tangente”, en tanto el vector  $\beta(v)$  es denominado vector normal y  $\gamma(v)$  es llamado “binormal”. Dichos vectores se relacionan de la manera siguiente:

$$1) \alpha(v) = \frac{d}{dv} f(v) \quad (5.20)$$

$$2) \beta(v) = \frac{d}{dv} \alpha(v)$$

$$3) \gamma(v) = \alpha(v) \times \beta(v)$$

Por otro lado, los vectores unitarios asociados con las expresiones (5.20) se describen a continuación:

$$1) \alpha(v)_u = \frac{\alpha(v)}{\|\alpha(v)\|} \quad (5.21)$$

$$2) \beta(v)_u = \frac{\beta(v)}{\|\beta(v)\|}$$

$$3) \gamma(v)_U = \frac{\gamma(v)}{\|\gamma(v)\|}$$

Considera ahora las relaciones siguientes:

$$1) \alpha(v)_U = n_3 \tag{5.22}$$

$$2) \beta(v)_U = n_2$$

$$3) \gamma(v)_U = n_1$$

Los vectores  $n_1$ ,  $n_2$  y  $n_3$  se muestran en la figura 5.9

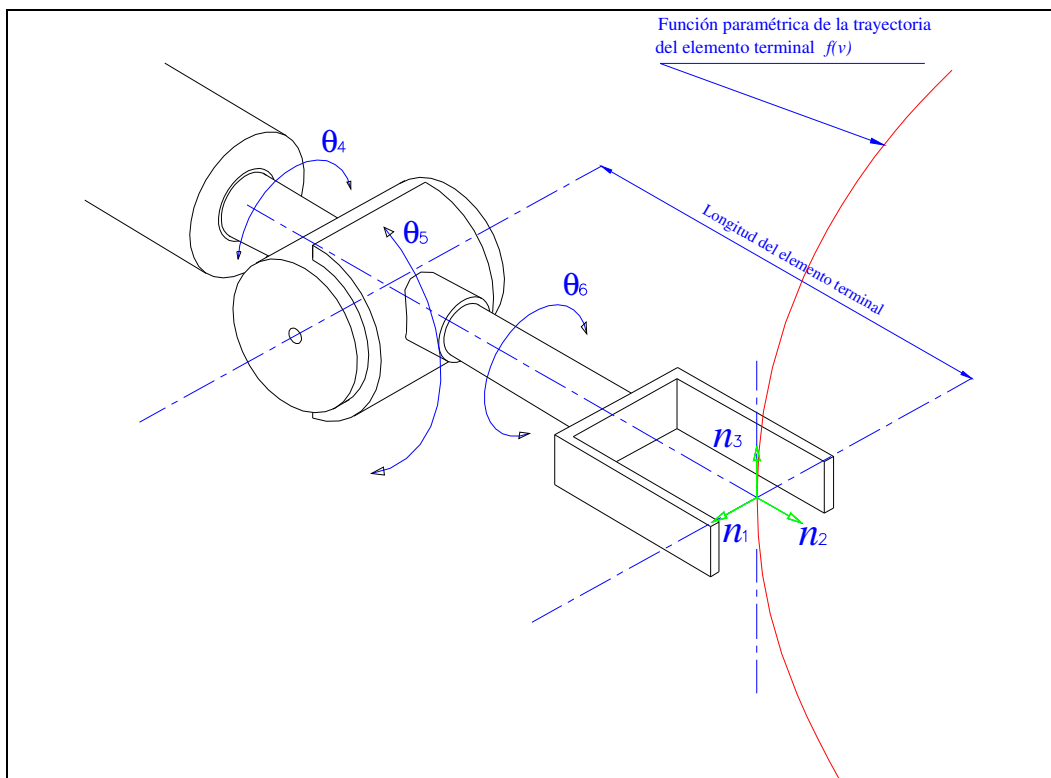


Figura 5.9 Vectores sobre el órgano terminal

### 5.6.3 Algunas relaciones importantes para la formulación del problema inverso

El método que será utilizado en este trabajo es el desacoplamiento cinemático el cual consiste en resolver por separado el problema para los tres primeros eslabones del robot;

esto es, para posición y, posteriormente, para la orientación. Considere la siguiente función:

$$fp(v) = f(v) - (d6 + d7)n2 \quad (5.23)$$

Dicha función es usada para modelar el movimiento de los tres primeros grados de libertad; esto, es una ecuación de posición. Nótese, por otro lado, que la trayectoria está compuesta, según la figura 5.6, por 3 partes:

- Una trayectoria lineal, desde la posición inicial (Home) a la posición inicial de la trayectoria helicoidal.
- Una trayectoria helicoidal, definida por la función paramétrica  $f(v)$ .
- Una trayectoria lineal, desde la posición final de la trayectoria helicoidal a la posición inicial (Home).

Por otro lado, la solución de la posición se determina según las expresiones siguientes:

$$\begin{aligned} px &= rp_1 \\ py &= rp_2 \\ pz &= rp_3 \end{aligned} \quad (5.24)$$

Donde  $px$ ,  $py$  y  $pz$  son los puntos definidos por la función  $fp(v)$ , por lo que el sistema a resolver para la parte de posición se compone de las siguientes ecuaciones:

$$\left\{ \begin{array}{l} px = rp_1, \\ py = rp_2, \\ pz = rp_3, \\ p0^2 + p3^2 = 1, \\ q0^2 + q1^2 = 1, \\ r0^2 + r1^2 = 1 \end{array} \right. \quad (5.25)$$

Nótese que los componentes de los Cuaterniones  $p$ ,  $q$  y  $r$  están asociados con los eslabones inicial, secundario y terciario, respectivamente. Es posible, por lo tanto

formular el problema. Nótese que el sistema de ecuaciones e incógnitas para el caso de la posición es cuadrado y, en consecuencia, soluble.

Para el caso de la parte de orientación, se tiene que dada la función paramétrica de trayectoria, se define la orientación del sistema de ejes  $e_j^6$ , localizado en el extremo final del robot (Ver figura 5.2), como:

$$\begin{aligned} e_1^6 &= n3 \\ e_2^6 &= n2 \\ e_3^6 &= n1 \end{aligned} \tag{5.26}$$

Obsérvese que el sistema de ecuaciones (5.26) tiene 9 ecuaciones y 6 incógnitas, por lo que el sistema es sobredeterminado y, por lo tanto, tiene múltiples soluciones. Para solventar este problema considérese la siguiente premisa:

*Si los valores encontrados de los Cuaterniones  $s$ ,  $t$  y  $u$  son exactos (correctos), entonces la diferencia entre las expresiones (5.25) son tales que:*

$$\begin{aligned} \mathbf{dif}_1 &= e_1^6 - n3 = 0 \\ \mathbf{dif}_2 &= e_2^6 - n2 = 0 \\ \mathbf{dif}_3 &= e_3^6 - n1 = 0 \end{aligned} \tag{5.27}$$

Aquí, las diferencias:  $\mathbf{dif}_1$ ,  $\mathbf{dif}_2$  y  $\mathbf{dif}_3$  deben de ser 0 para ser solución. Además,

$$\begin{aligned} dif_1 &= \sqrt{\mathbf{dif}_{1x}^2 + \mathbf{dif}_{1y}^2 + \mathbf{dif}_{1z}^2} \\ dif_2 &= \sqrt{\mathbf{dif}_{2x}^2 + \mathbf{dif}_{2y}^2 + \mathbf{dif}_{2z}^2} \\ dif_3 &= \sqrt{\mathbf{dif}_{3x}^2 + \mathbf{dif}_{3y}^2 + \mathbf{dif}_{3z}^2} \end{aligned} \tag{5.27}$$

Obsérvese que es posible definir el siguiente sistema de ecuaciones para el caso de la orientación:



$$\left\{ \begin{array}{l} dif_1 = 0 \\ dif_2 = 0 \\ dif_3 = 0 \\ s0^2 + s2^2 = 1 \\ t0^2 + t1^2 = 1 \\ u0^2 + u2^2 = 1 \end{array} \right\} \quad (5.28)$$

Por lo tanto, el sistema de ecuaciones e incógnitas en cuadrado y soluble para el problema de orientación.

#### 5.6.4 Formulación del problema cinemático inverso

El problema cinemático inverso relacionado con el robot PUMA 560 motivo de estudio se formula de la manera siguiente:

*Dados*

$$d_1, d_2, d_3, d_4, d_5, d_6 \in \mathfrak{R}, \alpha(\mathbf{v}), \beta(\mathbf{v}), \gamma(\mathbf{v}), f(x(\mathbf{v}), y(\mathbf{v}), z(\mathbf{v})) \in \mathfrak{R}^3, \text{ con}$$

$$\|\mathbf{p}\| = 1, \|\mathbf{q}\| = 1, \|\mathbf{r}\| = 1, \|\mathbf{s}\| = 1, \|\mathbf{t}\| = 1, \text{ y } \|\mathbf{u}\| = 1, \text{ encuentre}$$

$$\mathbf{p}, \mathbf{q}, \mathbf{r}, \mathbf{s}, \mathbf{t}, \mathbf{u} \in \mathfrak{R}^4 \text{ tal que:}$$

$$\left\{ \begin{array}{l} px = rp_x, \\ py = rp_y, \\ pz = rp, \\ p0^2 + p3^2 = 1, \\ q0^2 + q1^2 = 1, \\ r0^2 + r1^2 = 1 \end{array} \right\}$$

$$\left\{ \begin{array}{l} dif_1 = 0 \\ dif_2 = 0 \\ dif_3 = 0 \\ s0^2 + s2^2 = 1 \\ t0^2 + t1^2 = 1 \\ u0^2 + u2^2 = 1 \end{array} \right\}$$

Sean satisfechas.

### 5.6.5 Simulación de la cinemática inversa

En la tabla 5.3 se muestra la programación del problema inverso del robot motivo de estudio en este trabajo de tesis.

<p><i>FHome</i> : Función paramétrica que define la trayectoria de <i>Home</i> al punto Inicial de la trayectoria definida por <i>f</i></p> <p><i>FToHome</i> : Función paramétrica que define la trayectoria del punto final de la trayectoria definida por <i>f</i> a <i>Home</i></p> <p><i>PuntoInicial</i> : Punto inicial de la trayectoria</p> <p><i>PuntoFinal</i> : Punto final de la trayectoria</p> <p><i>f</i> : Función paramétrica de la función helicoidal con radio <i>Radio</i> y centro <i>Centro</i> en función de <i>v</i></p> <p><i>n3</i> : vector Tangente a la función <i>f</i></p> <p><i>n2</i> : vector normal a la trayectoria <i>f</i></p> <p><i>n1</i> : vector perpendicular a la normal y a la tangente de la función <i>f</i></p> <p><i>fp</i> : Función definida por <math>f \cdot (d6 + d8) \cdot n2</math> que determina la cinemática inversa de posición todo con respecto a <math>e_i</math></p>	<p>Funciones definidas para las trayectorias del órgano terminal, de <i>Home</i> a la posición inicial, la función de trayectoria y desde la posición final a <i>Home</i>.</p>
<p>ClearAll[v,f]</p>	
<p>Radio=150;</p>	<p>Radio del circulo</p>
<p>Centro={-100,400,600};</p>	<p>Centro del circulo para la definición de la trayectoria</p>
<p>f={Centro[[1]]+Radio Cos[v], Centro[[2]]+10 v, Centro[[3]]+Radio Sin[v]};</p>	<p>Función que define la trayectoria</p>
<p>n3=TangenteU[f,v];</p>	<p>Función unitaria tangente la función de la trayectoria.</p>
<p>n2=NormalU[f,v];</p>	<p>Función unitaria normal a función de la trayectoria.</p>

<code>n1=Cross[n2,n3];</code>	Función perpendicular a la función tangente y a la normal unitaria.
<code>fp=f - (d6+d8) n2;</code>	Función de la trayectoria de posición
<code>ClearAll[solp,arr,px,py,pz,p0,p3,q0,q1,r0,r1,s0,s2,t0,t1,u0,u2,v,solp,solq,solr,solucion];</code>	
<code>i=1;</code>	Inicialización del contador
<code>ValFinal=N[3 π];</code>	ValorFinal
<code>PuntoInicialP=fp/. v→0;</code>	PuntoInicial de la trayectoria de posición
<code>PuntoFinalP=fp/. V→ValFinal;</code>	PuntoFinal de la trayectoria de posición
<code>HomeP=rp/. {p0→1,q0→1,r0→1,p3→0,q1→0,r1→0};</code>	Home
<code>ValorInicial = 0;</code>	Valor Inicial
<code>Incremento=0.1;</code>	Incremento
<code>FHomeP=HomeP+(PuntoInicialP-HomeP) v;</code>	Trayectoria de Home a Punto Inicial.
<code>FToHomeP=PuntoFinalP+(HomeP-PuntoFinalP) v;</code>	Trayectoria del Punto Final a Home.
<code>varsp={p0,p3,q0,q1,r0,r1};</code>	VARIABLES de posición.
<code>varso={s0,s2,t0,t1,u0,u2};</code>	VARIABLES de orientación
<code>valip={0.99,0.001,0.999,0.001,0.999,0.001};</code>	Valores iniciales de posición.
<code>valio={0.835089978, 0.550113379,-0.660664546, 0.750681263, 1, 0};</code>	Valores iniciales de orientación.
<code>{is0,is2,it0,it1,iu0,iu2}={1,0,1,0,1,0};</code>	Inicializa los valores para utilizarlos en la

	resolución inicial del sistema de ecuaciones.
<b>Sistema de ecuaciones para el problema cinemática de posición</b>	
<p>EcuacionesP={rp[[1]]==px,  rp[[2]]==py,  rp[[3]]==pz,  p0<sup>2</sup>+p3<sup>2</sup>==1  q0<sup>2</sup>+q1<sup>2</sup>==1  r0<sup>2</sup>+r1<sup>2</sup>==1 }</p>	Sistema de ecuaciones de posición.
<b>Vector diferencia entre el vector del ultimo elemento terminal y la orientación de la función paramétrica</b>	
<p>dif1 = e16-n1//ExpandAll//Simplify;  dif2 = e26-n2//ExpandAll//Simplify;  dif3 = e36-n3//ExpandAll//Simplify;</p> <p>EcuacionesO={  Sqrt[dif1[[1]]<sup>2</sup>+dif1[[2]]<sup>2</sup>+dif1[[3]]<sup>2</sup>]=0,  Sqrt[dif2[[1]]<sup>2</sup>+dif2[[2]]<sup>2</sup>+dif2[[3]]<sup>2</sup>]=0,  Sqrt[dif3[[1]]<sup>2</sup>+dif3[[2]]<sup>2</sup>+dif3[[3]]<sup>2</sup>]=0,  s0<sup>2</sup>+s2<sup>2</sup>==1,  t0<sup>2</sup>+t1<sup>2</sup>==1,  u0<sup>2</sup>+u2<sup>2</sup>==1  };</p>	<p>Simplificación de la diferencia de los vectores unitarios para el problema de orientación</p> <p>Sistema de ecuaciones de orientación</p>
<b>Determinación de la cinemática inversa de posición para la trayectoria (FHomeP) de Home al punto inicial de la trayectoria helicoidal</b>	
<p>ClearAll[pz,p0,p3,q0,q1,r0,r1,s0,s2,t0,t1,u0,u2];</p> <p>For[m=ValorInicial,m&lt;=1,m+=Incremento,  ClearAll[px,py,pz,EcuacionesP];</p> <p>EcuacionesP={rp[[1]]==px,  rp[[2]]==py,  rp[[3]]==pz,  p0<sup>2</sup>+p3<sup>2</sup>==1  q0<sup>2</sup>+q1<sup>2</sup>==1  r0<sup>2</sup>+r1<sup>2</sup>==1 }</p> <p>SolucionPosicion=ERP[EcuacionesP,varsp,valip,FHomeP,v,If[m-Incremento&lt;ValorInicial,ValorInicial,m-Incremento],m,4];</p>	<p>Redefinición de las ecuaciones de posición.</p> <p>Encuentra la solución para el problema de</p>

<pre> valip= SolucionPosicion;solp[bb]= SolucionPosicion;bb++ ]; </pre>	<p>posición, para las ecuaciones EcuacionesPosicion, con un valor de <math>v = m</math></p>
	<p>se almacena la solución encontrada, misma que se utiliza como valor inicial para el siguiente valor de <math>v</math></p>
<b>Resolución de la cinemática inversa para la trayectoria helicoidal</b>	
<pre>ClearAll[px,py,pz];</pre>	
<pre>InOr=-bb;</pre>	<p>Contador para el inicio de la parte de orientación</p>
<pre>inc=Incremento;</pre>	<p>Incremento utilizado para el análisis.</p>
<pre>Vars={0.527,0.849,0.472,-0.881,0.082,0.996}; {is0, is2, it0, it1, iu0, iu2} = valio;</pre>	<p>Inicialización de las variables de posición y de orientación</p>
<pre>Varsp=varsp;</pre>	
<pre>cc=Simplify[EcuacionesP];</pre>	<p>Simplificación de las ecuaciones de posición</p>
<pre>For[nm=ValorInicial,nm→ValFinal,nm+=Incremento/2,</pre>	<p>Inicio del ciclo para la determinación de la cinemática inversa</p>
<pre>ClearAll[v,px,py,pz];</pre>	
<pre>dd=fp/.v→nm;</pre>	<p>Valor de <math>fp</math> cuando <math>v</math> vale <math>nm</math></p>
<pre>ClearAll[p0,p3,q0,q1,r0,r1];</pre>	
<pre>SolucionPosicion=ERP[cc, varsp, valip, dd, v, nm-Incremento, nm, 4];</pre>	<p>Determinando en</p>

<pre>valip=SolucionPosicion;</pre>	<p>primer lugar la parte de posición</p>
<pre>solp[bb]=SolucionPosicion;</pre>	<p>almacenando la solución</p>
<pre>bo=bb;</pre>	<p>y usandola como valor inicial para el siguiente ciclo</p>
<pre>ecs2 = EcuacionesO/.{v→nm,   p0→solp[bo][[1]],p3→solp[bo][[2]],   q0→solp[bo][[3]],q1→solp[bo][[4]],   r0→solp[bo][[5]],r1→solp[bo][[6]]//Simplify;</pre>	<p>Simplificación de las ecuaciones de orientación dada la solución de posición</p>
<pre>SolucionOrientacion=FindRoot[ecs2,{s0,is0},{s2,is2},{t0,it0},{t1,it1},{u0,iu0},   {u2,iu2},MaxIterations-&gt;1000];</pre>	<p>Determinación de la solución de la parte de orientación</p>
<pre>valio= {s0, s2, t0, t1, u0,u2}/.SolucionOrientacion;</pre>	<p>asignando la solución de orientación</p>
<pre>solso[bo]=valio ;</pre>	<p>para almacenarla</p>
<pre>{is0,is2,it0,it1,iu0,iu2}=valio;</pre>	<p>y utilizarla como punto de inicio del siguiente ciclo</p>
<pre>bb++;</pre>	<p>Contador del ciclo For</p>
<pre>];(*For*)</pre>	
<pre>FnOr=bb;</pre>	<p>Fin de la parte de orientación</p>
<pre>gf1=ParametricPlot3D[FHomeP,{v,0,1},DisplayFunction→Identity,Compiled→False];</pre>	<p>Gráfica de la función de Home al punto de inicio de la trayectoria helicoidal</p>
<pre>gf2=ParametricPlot3D[fp,{v,0,ValFinal},DisplayFunction→Identity,Compiled→False];</pre>	<p>Gráfica de la función que determina la posición de los</p>

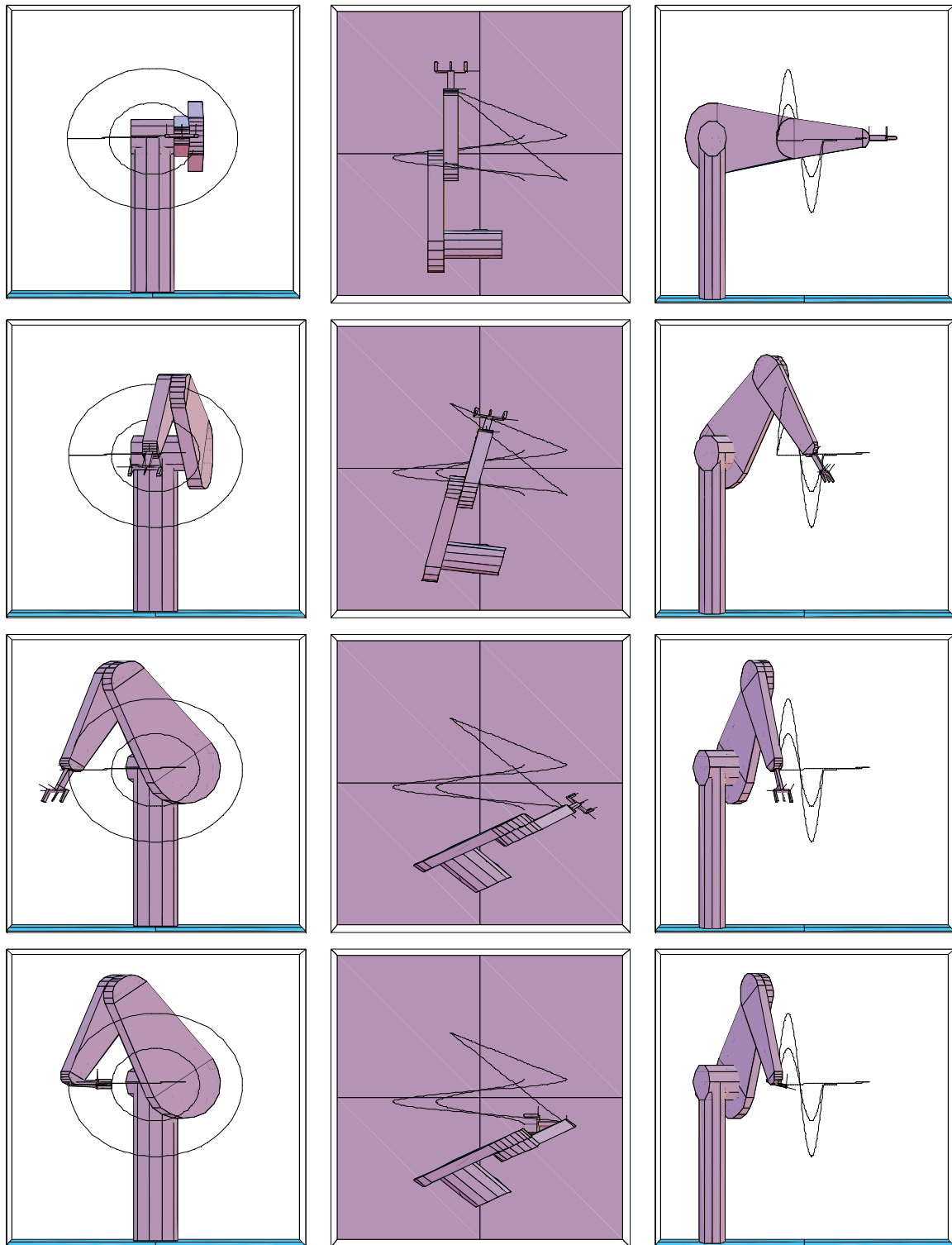
<pre>gf3=ParametricPlot3D[FToHomeP,{v,0,1},DisplayFunction-&gt;Identity,Compiled-&gt;False];  gf4=ParametricPlot3D[f,{v,0,ValFinal},DisplayFunction-&gt;Identity,Compiled-&gt;False];</pre>	<p>primeros 3 grados de libertad</p> <p>Gráfica de la función de la posición final de la trayectoria helicoidal a Home</p> <p>gráfica de la Trayectoria helicoidal</p>
<b>Graficación de la cinemática inversa desarrollada</b>	
<pre>For[j=1,j&lt;bb,j++, ClearAll[p0,p3,q0,q1,r0,r1,s0,s2,t0,t1,s0,s2];  p0=solp[j][[1]]; p3=solp[j][[2]]; q0=solp[j][[3]]; q1=solp[j][[4]]; r0=solp[j][[5]]; r1=solp[j][[6]];  If[j&gt;InOr ,If[j&lt;FnOr, s0=solso[j][[1]];s2=solso[j][[2]]; t0=solso[j][[3]];t1=solso[j][[4]]; u0=solso[j][[5]];u2=solso[j][[6]],  s0=1;s2=0;t0=1;t1=0;u0=1;u2=0], s0=1;s2=0;t0=1;t1=0;u0=1;u2=0];  Punto1={0,0,0}; Punto2=N[b1]; Punto3=N[Punto2+b2]; Punto4=N[Punto3+b3]; Punto5=N[Punto4+b4]; Punto6=N[Punto5+b5]; b7=L7 e16; b8=L8 e26;  Punto7=N[Punto6+b6]; Punto8=N[Punto7+b7]; Punto9=N[Punto7-b7]; Punto10=N[Punto8+b8]; Punto11=N[Punto9+b8]; Show[Superficie,gf1,gf2,gf3,gf4, Graphics3D[{</pre>	<p>Desde el primer punto de simulación hasta el final</p> <p>Se asignan los valores de los elementos de los Cuaterniones del arreglo de la solución para la parte de posición</p> <p>Al simular la trayectoria helicoidal, se utilizan los valores almacenados, de otra forma no se mueve el elemento terminal</p> <p>Puntos que definen los elementos del manipulador</p> <p>Vectores del elemento terminal</p> <p>Puntos que definen el elemento terminal</p> <p>Se muestra la</p>

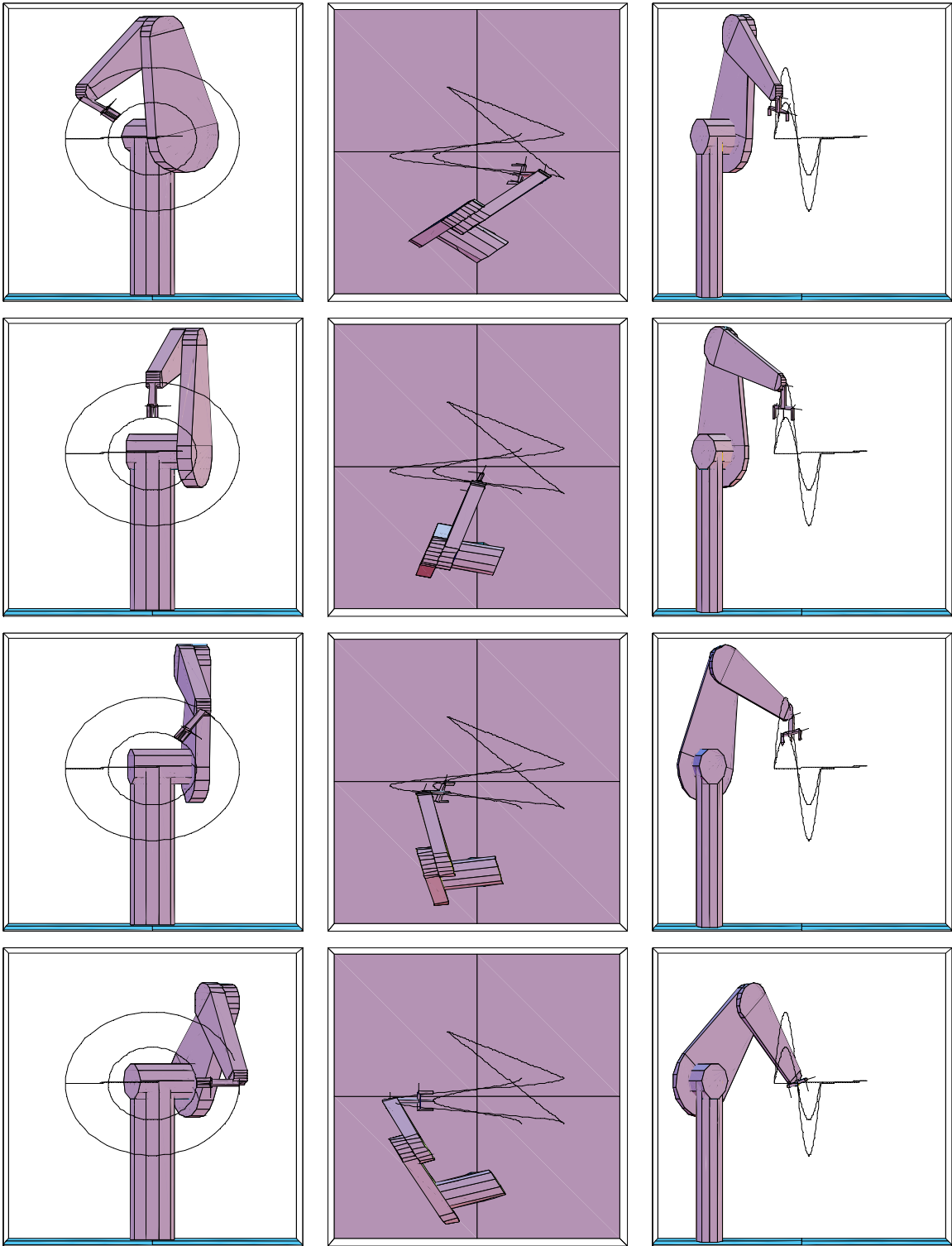
	superficie y las gráficas que definen las trayectorias
<pre> Cilindro[rad,L1], Cilindro[rad,rad+L2,-1,p,{rad,0,L1}]], Sardina[2.0 rad, 1.1 rad,anch,L3,1,Punto3,pq], Sardina[1.1 rad, 0.5rad,anch,L5-0.5 rad,1,Punto5,pr], Elemento[Punto6,Punto7,l4,1,e16,e26,e36], Elemento[Punto8,Punto9,l4/2,0,e16,e26,e36], Elemento[Punto8,Punto10,l4/2,1,e16,e26,e36], Elemento[Punto9,Punto11,l4/2,1,e16,e26,e36], PlotRange-&gt;{{-500,500},{-300,1300},{-5,1200}}, AspectRatio-&gt;Automatic,ViewPoint-&gt;{10.000,10.000, 10.0} </pre>	Elementos gráficos que definen los elementos del manipulador
	Rango de dibujo Punto de vista del observador

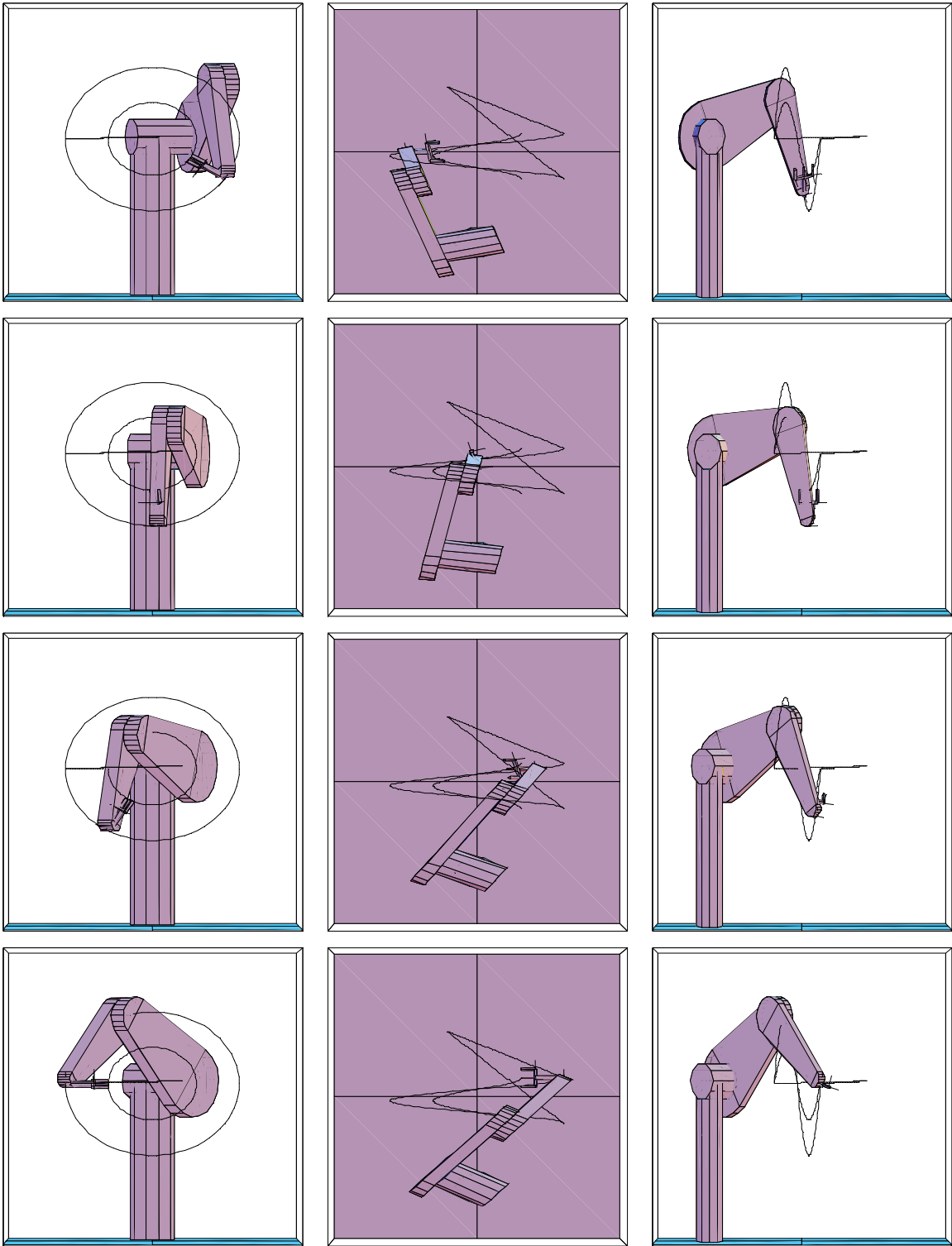
Tabla 5.3 Programación del problema cinemática inverso



Las figuras 5.10 muestran las salidas gráficas del problema inverso.







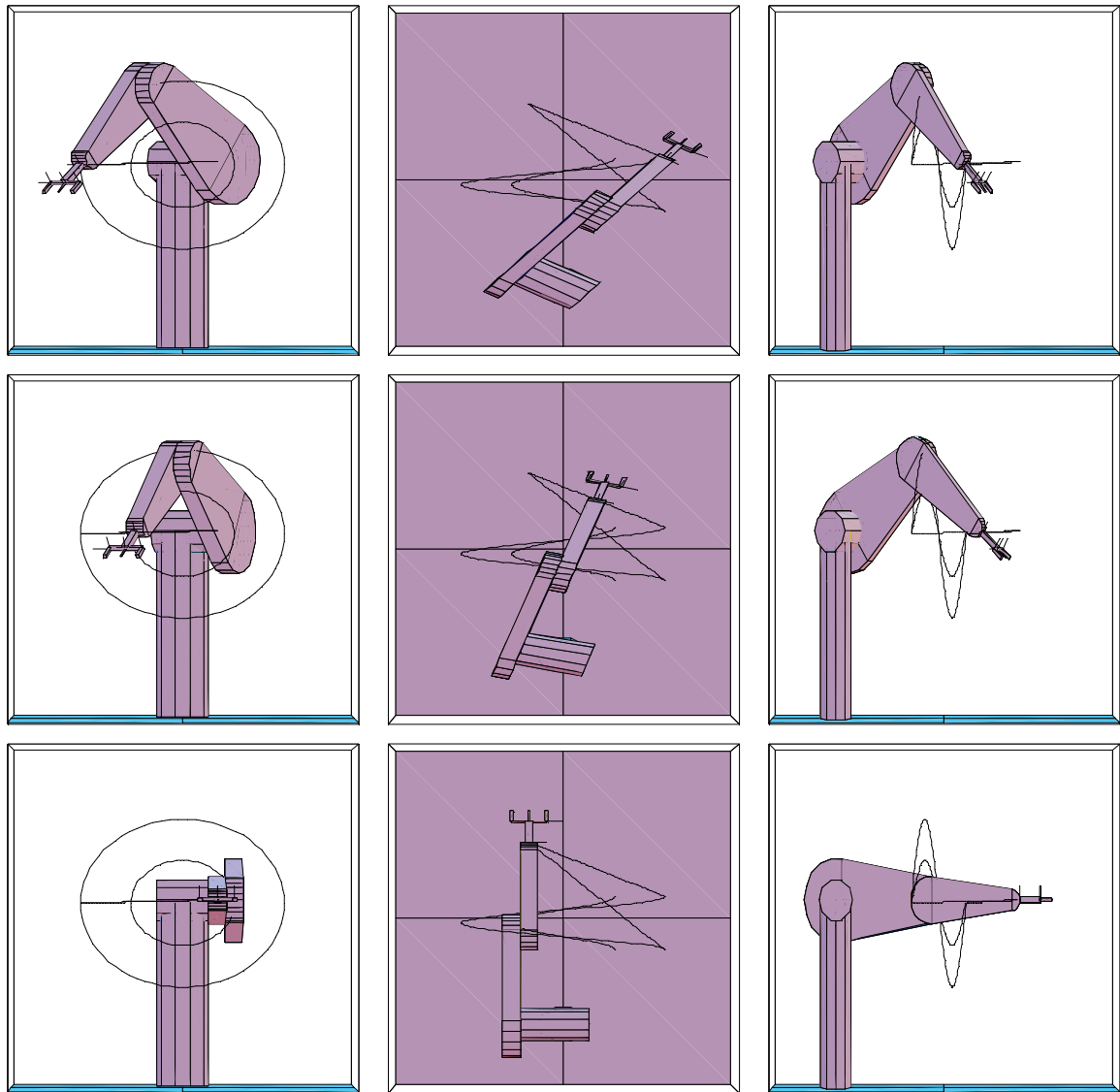


Figura 5.10 Salidas gráficas del problema inverso

## Capítulo 6

### Conclusiones

En este trabajo de tesis se ha modelado la cinemática de un robot industrial PUMA 560 utilizando el álgebra de Cuaterniones. Los objetivos descritos en la introducción de esta tesis fueron satisfechos. Esto es:

- Se modeló la cinemática inversa y directa de un robot PUMA 560 usando el álgebra de Cuaterniones y una trayectoria helicoidal.
- Se simplificaron las ecuaciones resultantes usando las propiedades del álgebra de Cuaterniones.
- Fueron programados y simulados los modelos obtenidos en Mathematica V5.

Los resultados y las conclusiones derivadas de esta tesis se resumen en los puntos siguientes:

- 1) El modelo de la cinemática inversa relacionada con el robot motivo de estudio en esta tesis generó un sistema de 12 ecuaciones algebraicas no lineales y 12 incógnitas del tipo polinomial.
- 2) El modelo de la cinemática directa relacionada con el robot motivo de estudio en esta tesis generó un sistema de 6 ecuaciones algebraicas no lineales y 6 incógnitas del tipo polinomial.
- 3) No existe un procedimiento general para resolver la cinemática inversa para cadenas cinemáticas de más de un grado de libertad por lo que se debe determinar un procedimiento universal.

- 4) La ecuación de la trayectoria en la cinemática inversa se da de forma paramétrica de forma tal que la orientación queda determinada automáticamente y puede colocarse otra función de la misma forma sin modificar el programa para su resolución.
- 5) El desacoplar el problema cinemática inverso simplifica su resolución.
- 6) Se requiere trabajar sobre la determinación de velocidades y aceleraciones para posteriormente en la dinámica del mismo.
- 7) La determinación de los valores iniciales cercanos a la solución del primer punto en la cinemática inversa de la función paramétrica tiene que estudiarse a detalle.
- 8) Se debe de trabajar sobre el modelo físico para reproducir la simulación generada en un robot PUMA.
- 9) El software de calculo formal Mathematica es una herramienta general que nos permite la modelación y simulación de forma sencilla y potente a la vez.

Finalmente, futuros trabajos por desarrollar relacionados con este trabajo de tesis son:

- Modelación dinámica del robot PUMA 560 usando Cuaterniones.
- Modelación completa cinemática y de trayectoria del robot PUMA 560
- Modelación de un robot industrial IRB 2400 usando una variante del álgebra de Cuaterniones.

## Referencias

- [1] Reyes L. Quaternions: Une Representation Parametrique Systematique Des Rotations Finies. Partie 1: Le Cadre Theorique. (1990). *Rapport de Recherche INRIA Rocquencourt*. France.
- [2] Reyes L. Quaternions: Une Representation Parametrique Systematique Des Rotations Finies. Partie II: Quelques applications. (1991). *Rapport de Recherche INRIA Rocquencourt*. France.
- [3] Cuenca J. Francisco,. “Modelado del análisis Cinemático de una Plataforma Paralela Plana. “: Una aplicación del álgebra de Quaterniones. Tesis de Grado de Maestro en Ingeniería Mecánica. (1997).
- [4] Jiménez E. Simulación de un proceso de manufactura con obstáculo en la línea de producción. México. (1998). (Tesis de Maestría en Ingeniería Mecánica). UNAM, Facultad de Ingeniería, División de Estudios de Posgrado, Sección Mecánica.
- [5] Martínez R. Modelación y simulación computacional de trayectorias tridimensionales México.(1997). Tesis (Maestría en Ingeniería Mecánica) UNAM, Facultad de Ingeniería, División de Estudios de Posgrado, Sección Mecánica.
- [6] Jiménez E., Reyes L.,.Modelación Cinemática De Posicionamiento De Un Robot De 3 Gdl Usando Quaterniones. Informe interno de investigación. (2004). ISBN 968-5844-01-1, Universidad La Salle Noroeste.
- [7] Méndez M. Modelado Cinemático de sistemas mecánicos de cadena cerrada México. (1995). Tesis (Maestría en Ingeniería Mecánica) UNAM, Facultad de Ingeniería, División de Estudios de Posgrado, Sección Mecánica.
- [8] Márquez M. Sobre los modelos dinámicos de robots de cadena abierta, parametrizados con Quaterniones (2000). Tesis (Doctorado en Ingeniería ) Universidad Anáhuac del Sur
- [9] Jiménez E., Reyes L., Pérez P., Ojeda P., Ruiz E., Domínguez G., Vázquez I. Sobre la parametrización de las rotaciones y reflexiones rígidas en el plano: aplicaciones al modelado de un robot delta. Informe interno de investigación 6.

- (2005). ISBN 968-5844-22-4, RED ALFA – Universidad La Salle Noroeste.
- [10] Hernández D. Síntesis para conducción de cuerpo rígido usando una rotación variante México. (2006). Tesis (Maestría en Ingeniería Mecánica) UNAM, Facultad de Ingeniería, División de Estudios de Posgrado, Sección Mecánica.
- [11] Jiménez E., Reyes L., Galindo F., Martínez A., Esquer H., Luna J., González I. Sistematización paso a paso de la modelación de trayectorias rectilíneas continuas. Informe interno de investigación (2004). Universidad la Salle Noroeste. ISBN 968-5844-04-6
- [12] John W. Gray. Mastering Mathematica, Programming Methods and Applications. 1994. Academic Press Inc. New. York.
- [13] Jiménez E., Cuenca F., Reyes L., Pérez A., Delfín J., Ruiz E., Ojeda E. Modelación cinemática de posicionamiento de un robot paralelo planar tipo rrr usando una números complejos.(2003). *Segundo Congreso de la Asociación Mexicana de Mecatrónica A.C.* Universidad Anáhuac del Sur. México D.F.
- [14] Hu Zhunqing, et al, KINEMATIC CONTROL OF SIX-JOINT MANIPULATORS AT SINGULAR CONFIGURATIONS, Power System and Communications Infrastructures, Beijing, September 2002
- [14] Olvera E., E. Jimenez, Reyes L. Modelado de un robot PUMA usando álgebra hipercompleja: Cuaterniones. Informe Interno de Investigación (2007. Impulsora de Desarrollo Dinámico S.A. de C.V. (ISBN en trámite). Cd. Obregón, Son., Mexico.