01173



DIVISION DE ESTUDIOS DE POSGRADO Facultad De Ingeniería

SIMULACION COMPUTACIONAL DE CADENAS CINEMATICAS ABIERTAS TRIDIMENSIONALES

MIGUEL ANGEL VEGA GONZALEZ

TESIS

PRESENTADA A LA DIVISION DE ESTUDIOS DE POSGRADO DE LA

FACULTAD DE INGENIERIA

DE LA

UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

COMO REQUISITO PARA OBTENER EL GRADO DE

MAESTRO EN INGENIERIA (MECANICA)

Director de Tesis: Dr. Luis Reyes Avila

CIUDAD UNIVERSITARIA, AGOSTO 1997.

TESIS CON FALLA DE ORIGEN



Universidad Nacional Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

A mis padres Miguel y Ma. Soledad, ya que ellos lograron darme las bases necesarias para poder siempre seguir avanzando.

A mi esposa Beatriz y mis dos hijas Blanca Angélica y Claudia Patricia por todo el apoyo que me dieron para iniciar y concluir esta etapa de mi formación profesional.

A todos mis hermanos que siempre me han alentado para seguir adelante en todo lo que me he propuesto.

Al Dr. Luis Reyes Avila que desde un principio logró infundirme confianza en que con el trabajo se pueden lograr grandes cosas.

A todos mis compañeros y maestros que de una u otra forma incidieron para llegar a este punto tan importante de mi vida.

CONTENIDO

	Pag.
	1
CAPITULO I 1. ALGUNAS DEFINICIONES BASICAS	2
CAPITULO II 2. ANTECEDENTES 2.1 ALGEBRA DE QUATERNIONES 2.2 REPRESENTACION PARAMETRICA DE ROTACIONES 2.3 REPRESENTACION PARAMETRICA DE REFLEXIONES	5 5 11 21
 CAPITULO III 3. MODELACION CINEMATICA 3.1 Modelación cinemática de un manipulador de tres grados de libertad, utilizando rotaciones con diferentes secuencias de 	34
movimientos	34 34
terminal en la configuración no deformada	41
deformada	41 42
terminal en la configuración deformada	50
deformada	50
utilizando rotaciones y reflexiones	52 53
terminal en la configuración deformada	57
deformada	57
libertad, utilizando rotaciones	59 59
3.3.2 Vectores que definen los eslabones y posicion del elemento terminal en la configuración no deformada 3.3.3 Ecuaciones explícitas optimizadas para la configuración no	62
deformada	64 65

3.3.6 Ecuaciones explicitas optimizadas para la configuración deformada 70 3.4 Modelación cinemática de un manipulador de cuatro grados de libertad tipo scara, utilizando rotaciones 71 3.4.1 Configuración no deformada 71 3.4.2 Vectores que definen los eslabones y posición del elemento terminal en la configuración no deformada 76 3.4.3 Ecuaciones explícitas optimizadas para la configuración no deformada 76 3.4.4 Configuración deformada 77 3.4.5 Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 82 CAPITULO IV 4 SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática du n robot de tres grados de libertad 83 4.1.2 Cinemática Directa 84 4.1.2 Cinemática Inversa 96 4.2 Cinemática du ur robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática linversa 120 4.3 Cinemática Inversa 120 <th>3.3.5 Vectores que definen los eslabones y posición del elemento</th> <th></th>	3.3.5 Vectores que definen los eslabones y posición del elemento	
deformada 70 3.4 Modelación cinemática de un manipulador de cuatro grados de libertad tipo scara, utilizando rotaciones 71 3.4.1 Configuración no deformada 71 3.4.2 Vectores que definen los eslabones y posición del elemento terminal en la configuración no deformada 76 3.4.3 Ecuaciones explicitas optimizadas para la configuración no deformada 76 3.4.4 Configuración deformada 77 3.4.5 Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada 76 3.4.5 Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada 71 3.4.6 Ecuaciones explicitas optimizadas para la configuración deformada 81 3.4.6 Ecuaciones explicitas optimizadas para la configuración deformada 82 CAPITULO IV 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1.1 Cinemática de un robot de tres grados de libertad 83 4.1.2 Cinemática Inversa 96 4.2.1 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 <t< td=""><td>3.3.6 Ecuaciones explícitas optimizadas para la configuración</td><td>19</td></t<>	3.3.6 Ecuaciones explícitas optimizadas para la configuración	19
3.4 Modelación cinematica de un manipuliador de cuatro grados de libertad tipo scara, utilizando rotaciones 71 3.4.1 Configuración no deformada 71 3.4.2 Vectores que definen los eslabones y posición del elemento terminal en la configuración no deformada 76 3.4.3 Ecuaciones explicitas optimizadas para la configuración no deformada 76 3.4.4 Configuración deformada 77 3.4.5 Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada 81 3.4.6 Ecuaciones explicitas optimizadas para la configuración deformada 81 3.4.6 Ecuaciones explicitas optimizadas para la configuración deformada 82 CAPITULO IV 4 SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática de robot de cuatro grados de libertad 106 4.2.2 Cinemática liversa 120 4.3 Cinemática de robot de cuatro grados de libertad 131 4.3.2 Cinemática de robot de cuatro grados de libertad 131 4.3.2 Cinemática inve	deformada	0
3.4.1 Configuración no deformada 71 3.4.2 Vectores que definen los eslabones y posición del elemento terminal en la configuración no deformada 76 3.4.3 Ecuaciones explicitas optimizadas para la configuración no deformada 76 3.4.4 Configuración deformada 77 3.4.5 Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada 81 3.4.6 Ecuaciones explicitas optimizadas para la configuración deformada 81 3.4.6 Ecuaciones explicitas optimizadas para la configuración deformada 82 CAPITULO IV 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1.1 Cinemática Directa 86 4.2 Cinemática Inversa 96 4.2 Cinemática Inversa 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Inversa 131 4.3.2 Cinemática Inversa 144 CAPITULO V 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 131 4.3.2 Cinemática Inversa 144 CAPITULO V 4.4 Evasión de obstáculos en la cinemática in	3.4 Modelacion cinematica de un manipulador de cuatro grados de libertad tipo scara, utilizando rotaciones 7	11
3.4.2 Vectores que definen los eslabones y posición del elemento terminal en la configuración no deformada 76 3.4.3 Ecuaciones explícitas optimizadas para la configuración no deformada 76 3.4.4 Configuración deformada 77 3.4.5 Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 82 <i>CAPITULO IV</i> 4 SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1.2 Cinemática Inversa 96 4.2 Cinemática Inversa 106 4.2.1 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Directa 131 4.3.1 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Inversa 131 4.3.2 Cinemática Inversa 131	3.4.1 Configuración no deformada	1
terminal en la configuración no deformada 76 3.4.3 Ecuaciones explícitas optimizadas para la configuración no deformada 76 3.4.4 Configuración deformada 77 3.4.5 Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 82 <i>CAPITULO IV</i> 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática Directa 84 4.1.2 Cinemática Inversa 96 4.2 Cinemática Inversa 96 4.2 Cinemática Inversa 106 4.2.1 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Inversa 131 4.3.2 Cinemática Inversa 144 <i>CAPITULO V</i> 14 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	3.4.2 Vectores que definen los eslabones y posición del elemento	
3.4.3 Ecuaciones explicitas optimizadas para la configuración no deformada 76 3.4.4 Configuración deformada 77 3.4.5 Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 82 CAPITULO IV 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1 Cinemática Inversa 96 4.2 Cinemática lorecta 106 4.2.1 Cinemática de un robot de cuatro grados de libertad 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 120 4.3 Cinemática lorecta 131 4.3.2 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.4 Evasión de obstáculos en la cinemáti	terminal en la configuración no deformada 7	6
deformada 76 3.4.4 Configuración deformada 77 3.4.5 Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 82 <i>CAPITULO IV</i> 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1.1 Cinemática Inversa 96 4.2 Cinemática Inversa 96 4.2.1 Cinemática de un robot tipo SCARA de cuatro grados de libertad 106 4.2.2 Cinemática Directa 106 4.3.1 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3.1 Cinemática Inversa 120 4.3 Cinemática Inversa 131 4.3.2 Cinemática Inversa 142 CAPITULO V 14 4.4 Evasión de obstáculos en la cinemátic	3.4.3 Ecuaciones explícitas optimizadas para la configuración no	
3.4.4 Configuración deformada 77 3.4.5 Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada 81 3.4.6 Ecuaciones explicitas optimizadas para la configuración deformada 82 CAPITULO IV 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1.1 Cinemática Directa 84 4.2.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Inversa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 131 4.3.2 Cinemática Inversa 144 CAPITULO V 14 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	deformada	'6
3.4.5 Vectores que definen los estabones y posición del elemento terminal en la configuración deformada. 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 82 CAPITULO IV 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1.1 Cinemática Directa 84 4.1.2 Cinemática loversa 96 4.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática loversa 120 4.3 Cinemática loversa 120 4.3 Cinemática loversa 120 4.3 Cinemática Directa 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 120 4.3 Cinemática Inversa 144 CAPITULO V 144 144 CAPITULO V 157 157 CONCLUSIONES 183 184 APENDICES 185	3.4.4 Configuración deformada 7	7
terminal en la configuración deformada 81 3.4.6 Ecuaciones explícitas optimizadas para la configuración deformada 82 CAPITULO IV 83 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1 Cinemática Directa 84 4.1.2 Cinemática Inversa 96 4.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3.2 Cinemática Inversa 120 4.3.2 Cinemática Inversa 120 4.3.2 Cinemática Inversa 131 4.3.2 Cinemática Inversa 144 CAPITULO V 14 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	3.4.5 Vectores que definen los eslabones y posición del elemento	
3.4.6 Ecuaciones explicitas optimizadas para la configuración deformada 82 CAPITULO IV 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1.1 Cinemática Directa 84 4.1.2 Cinemática Inversa 96 4.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.2 Cinemática Inversa 144 CAPITULO V 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 184 APENDICES 184 184	terminal en la configuración deformada	31
Gerormada 62 CAPITULO IV 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1.1 Cinemática Directa 84 4.1.2 Cinemática Inversa 96 4.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Directa 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática Inversa 144 CAPITULO V 131 4.3.2 Cinemática Inversa 144 CAPITULO V 144 CAPITULO V 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	3.4.6 Ecuaciones explicitas optimizadas para la configuración	10
CAPITULO IV 4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad 83 4.1.1 Cinemática Directa 84 4.1.2 Cinemática Inversa 96 4.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 144 CAPITULO V 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185 184		52
4. SIMULACION COMPUTACIONAL DE LA CINEMATICA 83 4.1 Cinemática de un robot de tres grados de libertad. 83 4.1.1 Cinemática Directa 84 4.1.2 Cinemática Inversa 96 4.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática Directa 106 4.2.2 Cinemática Directa 106 4.2.2 Cinemática loversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática loversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 144 CAPITULO V 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	CAPITULO IV	
4.1 Cinemática de un robot de tres grados de libertad. 83 4.1.1 Cinemática Directa 84 4.1.2 Cinemática Inversa 96 4.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática Inversa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 120 4.3 Cinemática Directa 131 4.3.2 Cinemática Inversa 144 CAPITULO V 144 CAPITULO V 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	4. SIMULACION COMPUTACIONAL DE LA CINEMATICA	33
4.1.1 Cinemática Directa 84 4.1.2 Cinemática Inversa 96 4.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 131 4.3.2 Cinemática Inversa 144 CAPITULO V 144 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	4.1 Cinemática de un robot de tres grados de libertad	33
4.1.2 Cinemática Inversa 96 4.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Directa 131 4.3.3 Cinemática Directa 131 4.3.4 Cinemática Inversa 144 CAPITULO V 144 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	4.1.1 Cinemática Directa	34
4.2 Cinemática de robot de cuatro grados de libertad 106 4.2.1 Cinemática Directa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 144 <i>CAPITULO V</i> 144 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	4.1.2 Cinemática Inversa 9) 6
4.2.1 Cinemática Directa 106 4.2.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 144 <i>CAPITULO V</i> 144 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	4.2 Cinemática de robot de cuatro grados de libertad 10)6
4.2.2 Cinemática Inversa 120 4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 144 CAPITULO V 144 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	4.2.1 Cinemática Directa 10)6
4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 131 4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 144 <i>CAPITULO V</i> 144 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	4.2.2 Cinemática Inversa 12	20
4.3.1 Cinemática Directa 131 4.3.2 Cinemática Inversa 144 <i>CAPITULO V</i> 144 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad 13	31
4.3.2 Cinematica Inversa 144 CAPITULO V 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad) 157 CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	4.3.1 Cinemática Directa 13	31
CAPITULO V 4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad). 157 CONCLUSIONES. 183 REFERENCIAS 184 APENDICES.		14
4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres grados de libertad). 157 CONCLUSIONES. 183 REFERENCIAS 184 APENDICES. 185	CAPITULO V	
grados de libertad). 157 CONCLUSIONES. 183 REFERENCIAS 184 APENDICES. 185	4.4 Evasión de obstáculos en la cinemática inversa (Robot de tres	
CONCLUSIONES 183 REFERENCIAS 184 APENDICES 185	grados de libertad)	57
REFERENCIAS	CONCLUSIONES	33
APENDICES	REFERENCIAS 18	34
	APENDICES	35

INTRODUCCION

En este trabajo se presenta la aplicación del álgebra de Quaterniones para generar las ecuaciones que gobiernan la cinemática de cadenas cinemáticas abiertas, las cuales se utilizan para generar programas en el lenguaje de cálculo formal Mathematica[®], que nos permitirán simular el movimiento de los manipuladores que se seleccionaron como ejemplos. El principal objetivo es demostrar la validez del álgebra de Quaterniones para la modelación de la cinemática de, básicamente, cualquier tipo de manipulador robótico con relativa facilidad. Lo anterior se logra al presentar la modelación de juntas rotacionales y prismáticas, que son las dos juntas básicas de la mayoría de los manipuladores, como se podrá ver en la clasificación que se presenta más adelante.

En el primer capítulo se dan algunas definiciones básicas dando una clasificación de los manipuladores robóticos de cadena abierta.

En el segundo capítulo se muestran las propiedades del álgebra de Quaterniones y se proponen dos transformaciones lineales (rotación y reflexión) realizando todas las demostraciones de las propiedades que les corresponden, además de demostrar cual es el significado de los parámetros que las caracterizan; muchas de las demostraciones son realizadas utilizando Mathematica.

En el tercer capítulo se realiza el proceso de modelación de tres manipuladores utilizando rotaciones, uno de tres grados de libertad, otro de cuatro y el último un tipo SCARA; durante el proceso de modelación del primer manipulador, se muestra el efecto de las secuencias de movimientos en el modelo final obtenido, el cual no es afectado, solamente se tienen diferentes puntos intermedios. El manipulador de tres grados es utilizado también para generar un modelo alterno al obtenido con puras rotaciones, dicho modelo se obtiene utilizando una combinación de rotaciones y reflexiones.

En el cuarto capítulo se muestra el código desarrollado en Mathematica para poder realizar las simulaciones computacionales de los tres manipuladores, los programas son acompañados por breves descripciones de las instrucciones utilizadas, estos programas permiten observar la forma que adquieren las ecuaciones cinemáticas explícitas de cada manipulador, ya que son generadas en los mismos.

Finalmente el quinto y último capítulo nos presenta una aplicación de los diferentes modelos que se pueden obtener al utilizar solamente rotaciones, reflexiones o una combinación de ellas. La aplicación es del manipulador de tres grados de libertad del cual se obtuvieron dos modelos cinemáticos diferentes en el capítulo III, y consiste en lograr la evasión de un obstáculo virtual en el proceso de cinemática inversa.

CAPITULO I

1. ALGUNAS DEFINICIONES BASICAS

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. 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 realizada naturalmente por un manipulador tipo SCARA. Los manipuladores pueden ser clasificados por muchos criterios, tales como su geometría o 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. Estos manipuladores son usualmente clasificados cinemáticamente sobre la base del primer brazo o las primeras 3 juntas, describiendo por separado el órgano terminal o muñeca. La mayoría de estos manipuladores caen en uno de los siguientes tipos de geometría, [5].

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

Manipuladores Articulados.- Estos también son llamados manipuladores antropomórficos. Un diseño común es el manipulador tipo codo, tal como el PUMA (Programmable Universal Manipulator for Assembly), en éste arreglo el eje de la junta z_3 es paralelo a z_2 y, z_2 y z_3 son perpendiculares a z_1 (Figura 1.1). Esta configuración proporciona, relativamente, una amplia libertad de movimiento en un espacio de trabajo compacto.



Figura 1.1 Manipulador articulado

Manipuladores Esféricos.- Este tipo de manipuladores se obtienen al remplazar la tercera junta de un manipulador antropomórfico por una junta prismática (Figura 1.2).



Figura 1.2 Manipulador esférico

Manipulador tipo SCARA.- El llamado SCARA (Selective Compliant Articulated Robert Assembly), mostrado en la Figura 1.3, es una configuración relativamente reciente que a incrementado su popularidad, el cual como su nombre lo índica está diseñado principalmente para operaciones de ensamble.



Figure 1.3 Manipulador tipo SCARA

Manipulador Cilíndrico.- La configuración de un manipulador cilíndrico se muestra en la Figura 1.4. Como se puede observar la primera junta es rotacional y produce una rotación alrededor de la base, mientras las otras dos juntas son prismáticas.



Figura 1.4 Manipulador cillndrico.

Manipulador Cartesiano.- Un manipulador cuyas primeras tres juntas sean prismáticas es conocido como manipulador cartesiano, en la Figura 1.5 se muestra dicha configuración.



Figura 1.5 Manipulador cartesiano

CAPITULO II

2. ANTECEDENTES

2.1. Algebra de Quaterniones.

Los Quaterniones se han utilizado en años recientes, entre otras cosas, para representar la forma paramétrica de rotaciones finitas de cuerpos rígidos, 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 apropladas de sus conceptos. En este trabajo, basándonos en [1].[2], se presentará una estructura algebraica mediante la cual sistematizaremos las aplicaciones estudiadas.

Definamos dos operaciones binarias sobre el conjunto \mathfrak{R}^4 , una operación aditiva $\oplus: \mathfrak{R}^4 \times \mathfrak{R}^4 \to \mathfrak{R}^4$ y otra multiplicativa $*: \mathfrak{R}^4 \times \mathfrak{R}^4 \to \mathfrak{R}^4$ mediante las cuales las parejas (\mathfrak{R}^4, \oplus) y $(\mathfrak{R}^4, *)$ forman, respectivamente, un grupo aditivo conmutativo y un grupo multiplicativo no conmutativo, definiremos además una multiplicación escalar $\bullet: \mathfrak{R} \times \mathfrak{R}^4 \to \mathfrak{R}^4$ y un producto interno $\langle \bullet, \bullet \rangle: \mathfrak{R}^4 \times \mathfrak{R}^4 \to \mathfrak{R}$ mediante el cual $\mathbf{Q} = (\mathfrak{R}^4, \oplus, *, \langle \bullet, \bullet \rangle)$ es un espacio vectorial provisto de un producto interno. Tal espacio vectorial será llamado espacio vectorial de Quaterniones. Consideremos el conjunto \mathfrak{R}^4 y definamos las operaciones siguientes:

i).-
$$(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}) \oplus (\alpha,\beta,\gamma,\delta) = (\mathbf{a} + \alpha, \mathbf{b} + \beta, \mathbf{c} + \gamma, \mathbf{d} + \delta)$$
 (2.1)

ii).-
$$(\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d})^*(\alpha,\beta,\gamma,\delta) = (\mathbf{a}\alpha - \mathbf{b}\beta - \mathbf{c}\gamma - \mathbf{d}\delta, \mathbf{a}\beta + \mathbf{b}\alpha + \mathbf{c}\delta - \mathbf{d}\gamma,$$
 (2.2)
 $\mathbf{a}\gamma - \mathbf{b}\delta + \mathbf{c}\alpha + \mathbf{d}\beta, \mathbf{a}\delta + \mathbf{b}\gamma - \mathbf{c}\beta + \mathbf{d}\alpha),$
 $\forall (\mathbf{a},\mathbf{b},\mathbf{c},\mathbf{d}), (\alpha,\beta,\gamma,\delta) \in \mathfrak{R}^4.$

La operación $\oplus : \mathfrak{R}^4 \times \mathfrak{R}^4 \to \mathfrak{R}^4$ es la suma usual en \mathfrak{R}^4 y se sabe que el conjunto (\mathfrak{R}^4, \oplus) es un grupo aditivo conmutativo. En [1] se estudiaron las propiedades de las operaciones $\oplus : \mathfrak{R}^4 \times \mathfrak{R}^4 \to \mathfrak{R}^4 \times \mathfrak{R}^4 \to \mathfrak{R}^4 \times \mathfrak{R}^4 \to \mathfrak{R}^4$, realizándose los desarrollos de manera manual. En esta parte concretaremos las demostraciones de manera simbólica mediante Mathematica[®], esto permitirá construir nuevas estructuras algebraicas para parametrizar las rotaciones e incluso reflexiones, simplificando los desarrollos algebraicos. Ahora mostraremos que las operaciones anteriores satisfacen las siguientes propiedades.

Teorema 2.1.1: La terna $\mathbf{Q} = (\mathfrak{R}^4, \oplus, *)$ es un campo no conmutativo.

Demostración: Sabemos que (\Re^4, \oplus) es un grupo aditivo conmutativo, por lo que solo falta mostrar que $(\Re^4, *)$ es un grupo multiplicativo no conmutativo. Para ello debe cumplir las propiedades siguientes:

i).- La operación *: ℜ⁴ × ℜ⁴ → ℜ⁴ es asociativa. En decir, siendo p,q,s ∈ Q, donde p={a,b,c,d}, q={e,f,g,h}, s={w,x,y,z}, tenemos que: p°(q°s)={p°q}°s. (Las demostraciones son realizadas con Mathematica[®] y serán encerradas en un marco como se muestra a continuación).

In[1]:= (Este símbolo significa entrada de datos u operaciones y aparece automaticamente al correr el programa) (* Se define el producto de dos Quaterniones ProductoQ[p,q] *) (comentarios) $\label{eq:productoQ[p_q_]:= p[[1]] * q[[1]] * p[[2]] * q[[2]] * q[[3]] * q[[3]] * p[[4]] * q[[4]], }$ $\begin{array}{l} p[[1]] * q[[2]] * p[[2]] * q[[1]] * p[[3]] * q[[4]] * p[[4]] * q[[3]]. \\ p[[1]] * q[[3]] * p[[2]] * q[[4]] * p[[3]] * q[[1]] * p[[4]] * q[[2]]. \end{array}$ p[[1]] * q[[4]] + p[[2]] * q[[3]] - p[[3]] * q[[2]] + p[[4]] * q[[1]]); In[3]:= (* se asignan componentes a los Quaterniones a utilizar en la demostración, el punto y coma sirve para evitar que se despliegue el resultado de la operación *) $p = \{a, b, c, d\}; q = \{e, f, g, h\}; s = \{w, x, y, z\};$ In[5]:= (* Se demuestra la asociatividad en la multiplicación, es decir que "p*(q*s)=(p*q)*s", se utiliza el signo === para comparar las dos expresiones y se utiliza Expand[] para que pueda realizarse la comparación *) Expand[ProductoQ[p,ProductoQ[q,s]]] === Expand[ProductoQ[ProductoQ[p,q],s]] Out[6]= (Este simbolo significa salida de datos o resultados y se genera automaticamente) True

Podemos ver que Mathematica[®] nos da como salida de la comparación la palabra **True**, lo que significa que las expresiones comparadas son iguales, es decir la multiplicación definida si es *asoclativa*.

- ii).- El elemento 1=(1,0,0,0) ∈ Q es tal que 1*p = p*1 = p, ∀p ∈ Q, y es llamado elemento neutro para la multiplicación en Q.
- iii).- ∀p ∈ Q, p ≠ (0,0,0,0), existe p' ∈ Q, tal que p*p' = 1. el elemento p' es conocido como el inverso multiplicativo de p. Si tenemos que p={a,b,c,d} y p'={e,f,g,h} ∈ Q tal que p*p' = 1, se tiene que:

$$p' = \left\{ \frac{a}{a^2 + b^2 + c^2 + d^2}, \frac{-b}{a^2 + b^2 + c^2 + d^2}, \frac{-c}{a^2 + b^2 + c^2 + d^2}, \frac{-d}{a^2 + b^2 + c^2 + d^2} \right\}$$

```
in[1]:=
   (* Se definen los Quaterniones p y p' *)
   p={a,b,c,d}; p'={e,f,g,h};
in[3]:=
   (* como p^*p' = (1,0,0,0), se tiene un conjunto de 4 ecs*)
   pp'=ProductoQ[p,p']; (* Se calcula el producto p*p' *)
   ec1=pp'[[1]]==1
   ec2=pp'[[2]]==0
   ec3=pp'[[3]]==0
   ec4=pp'[[4]]==0
Out[5]=
   a e - b f - c q - d h == 1
Out[6]=
   be + af - dq + ch == 0
Out[7]=
   ce + d f + a g - b h == 0
Out[8]=
   de - cf + bg + a h == 0
in[9]:=
   (* Se resuelven las ecuaciones utilizando la función Solve[] *)
   Solve[{ec1,ec2,ec3,ec4},{e,f,g,h}]//Simplify
Out[11]=
                                      ъ
              а
{{e -> ----->, f -> -(----->),
      С
                                       d
  g -> -(-----), h -> -(-----)})
```

La solución del sistema de ecuaciones es el que se muestra en la salida Out[11], que corresponde a las componentes de p', es decir:

$$\mathbf{p}^{*} = \left\{ \frac{\mathbf{a}}{\mathbf{a}^{2} + \mathbf{b}^{2} + \mathbf{c}^{2} + \mathbf{d}^{2}}, \frac{-\mathbf{b}}{\mathbf{a}^{2} + \mathbf{b}^{2} + \mathbf{c}^{2} + \mathbf{d}^{2}}, \frac{-\mathbf{d}}{\mathbf{a}^{2} + \mathbf{b}^{2} + \mathbf{c}^{2} + \mathbf{d}^{2}}, \frac{-\mathbf{d}}{\mathbf{a}^{2} + \mathbf{b}^{2} + \mathbf{c}^{2} + \mathbf{d}^{2}} \right\}$$
(2.3)

iv).- La operación $*: \mathfrak{R}^4 \times \mathfrak{R}^4 \to \mathfrak{R}^4$ es no conmutativa, es decir, $\forall p,q \in Q, p^*q \neq q^*p$.

```
In[1]:=
    (* Se define el conjugado de un Quaternion *)
    ConjugadoQ[p_]:={p[[1]],-p[[2]],-p[[3]],-p[[4]]};
In[3]:=
    (* Se definen Quaterniones p y q *)
    p={a,b,c,d}; q={e,f,g,h};
In[5]:=
    (* Se obtienen los producto p*q y q*p *)
    pq=ProductoQ[p,q];
    qp=ProductoQ[q,p];
In[8]:=
    (* Se comparan los productos *)
    pq === qp
Out[9]=
    False
```

De la demostración anterior se determinó que la multiplicación definida es no conmutativa, ya que $\forall p,q \in Q, p^*q \neq q^*p$.

v).- Las propiedades siguientes de distributividad deben ser son satisfechas:

$$(p \oplus q)^* s = p^* s \oplus q^* s, \qquad (2.4)$$
$$p^*(q \oplus s) = p^* q \oplus p^* s,$$

Demostración: En efecto,

```
in[1]:=
    (* Se definen los Quaterniones p,q,s *)
    p={a,b,c,d}; q={e,f,g,h}; s={w,x,y,z};
in[3]:=
    (* Se comprueba la igualdad de las propiedades de distributividad *)
    Expand[ProductoQ[(p+q),s]] === Expand[ProductoQ[p,s]+ProductoQ[q,s]]
    Expand[ProductoQ[p,(q+s)]] === Expand[ProductoQ[p,q]+ProductoQ[p,s]]
Out[4]=
    True
Out[5]=
    True
```

8

Es bien conocido que la operación $\bullet: \mathfrak{R} \times \mathbf{Q} \to \mathbf{Q}$ definida $\forall \alpha \in \mathfrak{R}$, por :

$$\alpha \bullet (\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) = (\alpha \mathbf{a}, \alpha \mathbf{b}, \alpha \mathbf{c}, \alpha \mathbf{d}), \quad \forall (\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}) \in \mathfrak{R}^{\bullet}$$
(2.5)

es una multiplicación escalar y por consecuencia Q es un espacio vectorial. Consideremos los subespacios siguientes de Q:

$$Q_{R} = \{(a,0,0,0) : a \in \Re\} \subset Q,$$
 (2.6)

$$Q_v = \{(0,b,c,d) : b,c,d \in \Re\} \subset Q,$$
 (2.7)

los cuales son isomórficos a \Re y \Re^3 respectivamente, puesto que las transformaciones $T_R: Q_R \rightarrow \Re$, $T_V: Q_V \rightarrow \Re^3$, definidas por:

$$T_{R}(a,0,0,0) = a, \quad \forall (a,0,0,0) \in Q_{R}$$
 (2.8)

$$T_{V}(0,b,c,d) = (b,c,d), \quad \forall (0,b,c,d) \in Q_{V}$$
 (2.9)

son isomorfismos.

Con la operación escalar definida, podemos representar el inverso multiplicativo de p={a,b,c,d}, obtenido anteriormente como p':

$$p' = \frac{1}{a^2 + b^2 + c^2 + d^2} \cdot \{a, -b, -c, -d\} = \frac{1}{a^2 + b^2 + c^2 + d^2} \cdot \overline{p} . \quad (2.10)$$

Donde $\overline{p} = \{a, -b, -c, -d\}$ lo llamaremos el conjugado de $p = \{a, b, c, d\}$. Con lo anterior podemos representar el elemento $p \in Q$ de la forma siguiente:

$$\mathbf{p} = \mathbf{p}_{\mathbf{R}} \oplus \mathbf{p}_{\mathbf{V}}, \quad \mathbf{p}_{\mathbf{R}} \in \mathbf{Q}_{\mathbf{R}}, \quad \mathbf{p}_{\mathbf{V}} \in \mathbf{Q}_{\mathbf{V}}$$
(2.11)

De acuerdo con los isomorfismos T_R : $Q_R \rightarrow \Re$, T_V : $Q_V \rightarrow \Re^3$, podemos decir que p_R y p_V son respectivamente la parte real y la parte vectorial de $p \in Q$. Además el conjugado satisface los resultados siguientes.

Teorema 2.1.2: Si $p = (p_0, p_1, p_2, p_3) \in Q$ y $q = (q_0, q_1, q_2, q_3) \in Q$, se tiene que:

i).-
$$\overline{p \oplus q} = \overline{p} \oplus \overline{q}$$
,
ii).- $\overline{p^*q} = \overline{p}^*\overline{q}$,
iii).- $p^*\overline{p} = \overline{p}^*p \in Q_R$. (2.12)

Demostración: Como lo hicimos anteriormente la demostración se realizará utilizando Mathematica[®]. En efecto:

```
in[1]:=
     (* Se define el conjugado de un Quaternión *)
     ConjugadoQ[p_]:={p[[1]], - p[[2]], - p[[3]], - p[[4]]};
In[3]:=
    (* Se definen Quaterniones p y q *)
     p = \{a, b, c, d\}; q = \{e, f, g, h\};
In[5]:=
    (* iniciso i) *)
     ConjugadoQ[p+q] == ConjugadoQ[p]+ConjugadoQ[q]
Out[6]=
     True
in[7]:=
    (* iniciso ii) *)
     ConjugadoQ[ProductoQ[p,q]] == ProductoQ[ConjugadoQ[q],ConjugadoQ[p]]
Out[8]=
     True
in[9]:=
    (* iniciso iii) *)
    ProductoQ[p,ConjugadoQ[p]] == ProductoQ[ConjugadoQ[p],p]
Out[10]=
    True
```

Es bien conocido que la transformación $\langle \bullet, \bullet \rangle$: $\mathfrak{R}^4 \times \mathfrak{R}^4 \to \mathfrak{R}$ definida por $\langle \mathbf{p}, \mathbf{q} \rangle = \mathbf{p}_0 \mathbf{q}_0 + \mathbf{p}_1 \mathbf{q}_1 + \mathbf{p}_2 \mathbf{q}_2 + \mathbf{p}_3 \mathbf{q}_3$ es un producto interno en \mathfrak{R}^4 por lo que $(\mathbf{Q}, \oplus, \bullet, \langle \bullet, \bullet \rangle)$ es un espacio vectorial con producto interno. Además se tiene que la función $\| \bullet \| : \mathfrak{R}^4 \to \mathfrak{R}$, definida por $\| \mathbf{p} \| = \langle \mathbf{p}, \mathbf{p} \rangle^{1/2}$, es la norma para el producto interno, con lo que podemos concluir que $(\mathbf{Q}, \oplus, \bullet, \| \bullet \|)$ es un espacio normado. Un resultado importante para nuestra álgebra se presenta en el siguiente teorema.

Teorema 2.1.3: La transformación $\langle \bullet, \bullet \rangle$: $\mathfrak{R}^4 \times \mathfrak{R}^4 \to \mathfrak{R}$ puede representarse de la forma siguiente:

$$\langle \mathbf{p}, \mathbf{q} \rangle \doteq \frac{1}{2} \mathbf{T}_{\mathbf{R}} (\overline{\mathbf{p}}^{*} \mathbf{q} \oplus \overline{\mathbf{q}}^{*} \mathbf{p}), \quad \forall \mathbf{p}, \mathbf{q} \in \mathbf{Q}$$

```
In[1]:=
    (* Se define el conjugado de un Quaternion *)
    ConjugadoQ[p_]:={p[[1]].-p[[2]].-p[[3]].-p[[4]]};
In[3]:=
    (* Se definen Quaterniones p y q *)
    p={p0,p1,p2,p3}; q={q0,q1,q2,q3};
In[5]:=
    (* Se determina el producto de la definición (p * q ⊕ q * p) *)
    ProductoQ[ConjugadoQ[p],q]+ProductoQ[ConjugadoQ[q],p]
Out[6]=
    {2 p0 q0 + 2 p1 q1 + 2 p2 q2 + 2 p3 q3, 0, 0, 0}
```

De lo anterior podemos ver que $T_R(\overline{p}^*q \oplus \overline{q}^*p) = 2(p_0q_0 + p_1q_1 + p_2q_2 + p_3q_3) = 2(p,q).$

2.2. Representación paramétrica de rotaciones.

En esta parte utilizaremos el álgebra de Quaterniones, presentada anteriormente, para presentar de forma paramétrica una rotación de cuerpo rígido. Lo anterior se realizará mediante a definición de una transformación lineal $\rho(p, \bullet)$: $Q \rightarrow Q$. Se mostraran algunas de sus propiedades, además de la matriz asociada a la transformación respecto a la base canónica de \Re^4 , que se trata de una matriz 4x4, si la transformación presentada se restringe a Q_v , es decir, $\rho(p, \bullet)$: $Q_v \rightarrow Q_v$, se podrá obtener fácilmente la matriz 3x3 que se utiliza comúnmente para realizar las rotaciones de cuerpo rígido. Definamos la siguiente transformación lineal $\rho(p, \bullet)$: $Q \rightarrow Q$, con $p \in Q$ fijo, como:

$$\rho(\mathbf{p},\mathbf{q}) = \mathbf{p}^* \mathbf{q}^* \mathbf{p}^{-1} = \frac{1}{\|\mathbf{p}\|^2} \cdot (\mathbf{p}^* \mathbf{q}^* \overline{\mathbf{p}}), \quad \forall \mathbf{q} \in \mathbf{Q} , \qquad (2.13)$$

Algunas de las propiedades desarrolladas anteriormente son utilizadas en los teoremas siguientes referentes a la transformación anterior, la cual representa una *rotación*.

Teorema 2.2.1: La transformación $\rho(\mathbf{p}, \bullet)$: $\mathbf{Q} \rightarrow \mathbf{Q}$ es lineal, ortogonal y $\rho(\mathbf{p}, \mathbf{q}) \in \mathbf{Q}_{V}$. $\forall \mathbf{q} \in \mathbf{Q}_{V}$.

Demostración: Para demostrar al linealidad de la transformación, utilizamos las propiedades de asociatividad y distributividad de la estructura algebraica de Q.

ANTECEDENTES

$$\rho(\mathbf{p}, (\mathbf{q} \oplus \mathbf{s})) = \frac{1}{\|\mathbf{p}\|^2} \cdot \{\mathbf{p}^* (\mathbf{q} \oplus \mathbf{s})^* \overline{\mathbf{p}}\} = \frac{1}{\|\mathbf{p}\|^2} \cdot \{\mathbf{p}^* (\mathbf{q}^* \overline{\mathbf{p}} \oplus \mathbf{s}^* \overline{\mathbf{p}})\} = \frac{1}{\|\mathbf{p}\|^2} \cdot \{\mathbf{p}^* \mathbf{q}^* \overline{\mathbf{p}} \oplus \mathbf{p}^* \mathbf{s}^* \overline{\mathbf{p}}\} = \rho(\mathbf{p}, \mathbf{q}) \oplus \rho(\mathbf{p}, \mathbf{s}), \quad \forall \mathbf{p}, \mathbf{q}, \mathbf{s} \in \mathbf{Q}$$

У

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

Con lo anterior se demuestra la linealidad de la transformación, es decir que $\rho(\mathbf{p}, \mathbf{e}) \in \mathbf{L}(\mathbf{Q}, \mathbf{Q})$. Ahora para la demostración de la ortogonalidad de $\rho(\mathbf{p}, \mathbf{e})$ utilizamos el Teorema 2.1.2 y el Teorema 2.1.3. Para que la transformación sea ortogonal debe cumplirse que:

$$\langle \rho(\mathbf{p},\mathbf{q}), \rho(\mathbf{p},\mathbf{s}) \rangle = \langle \mathbf{p},\mathbf{s} \rangle \quad \forall \mathbf{q},\mathbf{s} \in \mathbf{Q}.$$

$$\langle \rho(\mathbf{p},\mathbf{q}),\rho(\mathbf{p},\mathbf{s})\rangle = \frac{1}{2\|\mathbf{p}\|^4} T_R \{ (\overline{\mathbf{p}^*\mathbf{q}^*\overline{\mathbf{p}}})^* (\mathbf{p}^*\mathbf{s}^*\overline{\mathbf{p}}) \oplus (\overline{\mathbf{p}^*\mathbf{s}^*\overline{\mathbf{p}}})^* (\mathbf{p}^*\mathbf{q}^*\overline{\mathbf{p}}) \}$$

$$= \frac{1}{2\|\mathbf{p}\|^4} T_R \{ (\mathbf{p}^*\overline{\mathbf{q}}^*\overline{\mathbf{p}})^* (\mathbf{p}^*\mathbf{s}^*\overline{\mathbf{p}}) \oplus (\mathbf{p}^*\overline{\mathbf{s}}^*\overline{\mathbf{p}})^* (\mathbf{p}^*\mathbf{q}^*\overline{\mathbf{p}}) \}$$

$$= \frac{1}{2\|\mathbf{p}\|^2} T_R \{ (\mathbf{p}^*\overline{\mathbf{q}}^*\mathbf{s}^*\overline{\mathbf{p}}) \oplus (\mathbf{p}^*\overline{\mathbf{s}}^*\mathbf{q}^*\overline{\mathbf{p}}) \} = \frac{1}{2\|\mathbf{p}\|^2} T_R \{ (\mathbf{p}^*(\overline{\mathbf{q}}^*\mathbf{s} \oplus \overline{\mathbf{s}}^*\mathbf{q})^*\overline{\mathbf{p}}) \}$$

$$= \frac{1}{2\|\mathbf{p}\|^2} T_R \{ (\mathbf{p}^*2\langle \mathbf{q},\mathbf{s}\rangle^*\overline{\mathbf{p}}) \} = \frac{2\langle \mathbf{q},\mathbf{s}\rangle}{2\|\mathbf{p}\|^2} T_R \{ (\mathbf{p}^*\overline{\mathbf{p}}) \} = \langle \mathbf{q},\mathbf{s}\rangle$$

Finalmente observemos que si q = {0, q₁, q₂, q₃} $\in Q_v$, entonces $p(p,q) \in Q_v$. En Efecto

In[1]:=

(* Se define la transformación lineal p(p,q) y utilizando las definiciones hechas anteriormente en Mathematica *)

RotacionQ[p_,q_]:= (1/np2)*ProductoQ[p,ProductoQ[q,Con]ugadoQ[p]]];

```
(* np2= norma de p al cuadrado *)
```

In[3]:=

(* Se definen Quaterniones p y q *)

p={p0,p1,p2,p3}; q={0,q1,q2,q3};

In[5]:=

(* Se determina $\rho(p,q)$ *)

Collect[RotacionQ[p,q],{np2,q1,q2,q3}]

```
Out[6]

\frac{2 2 2 2 2}{(p^{0} + p^{1} - p^{2} - p^{3})q^{1} + (2 p^{1} p^{2} - 2 p^{0} p^{3})q^{2} + (2 p^{0} p^{2} + 2 p^{1} p^{3})q^{3}}{np^{2}},

\frac{(2 p^{1} p^{2} + 2 p^{0} p^{3})q^{1} + (p^{0} - p^{1} + p^{2} - p^{3})q^{2} + (-2 p^{0} p^{1} + 2 p^{2} p^{3})q^{3}}{np^{2}},

\frac{(-2 p^{0} p^{2} + 2 p^{1} p^{3})q^{1} + (2 p^{0} p^{1} + 2 p^{2} p^{3})q^{2} + (p^{0} - p^{1} - p^{2} + p^{3})q^{3}}{np^{2}},
```

Teorema 2.2.2: La transformación p(p,•) es una rotación.

Demostración: Algunas de las propiedades necesarias para que la transformación sea una rotación se han demostrado ya en el Teorema 2.2.1, por ello solo resta demostrar que el determinante de la matriz asociada a la transformación $M_{\rho(p,e)} \in M_{4x4}$ es +1. En efecto,

```
In[1]:=
      (* Se define el Quaternión p *)
     p = \{p0, p1, p2, p3\};
     (* se define la base canónica de R<sup>4</sup>*)
     e1 = \{1,0,0,0\}; e2 = \{0,1,0,0\}; e3 = \{0,0,1,0\}; e4 = \{0,0,0,1\};
In[5]:=
     (* Se determina la matriz de la transformación (MRot) *)
     (* Se inicializa la lista Mrot donde se almacenará la matriz *)
     MRot={(),{),{},{},{}};
     (* Se aplica la rotación a la base canónica *)
     e11 = RotacionQ[p,e1];
     e21 = RotacionQ[p,e2];
     e31 = RotacionQ[p.e3];
     e41 = RotacionQ[p,e4];
     (* Se determinan los elementos de la matriz *)
     MRot = Insert[MRot,e11[[1]],{1,1}];
     MRot = Insert[MRot,e11[[2]],{2,1}];
     MRot = Insert[MRot,e11[[3]],{3,1}];
     MRot = Insert[MRot,e11[[4]],{4,1}];
     MRot = Insert[MRot,e21[[1]],{1,2}];
```

MRot = Insert[MRot,e21[[2]].{2.2}]; MRot = Insert[MRot,e21[[3]],{3.2}]; MRot = Insert[MRot,e21[[4]],{4,2}]; MRot = Insert[MRot,e31[[1]],{1,3}]; MRot = Insert[MRot,e31[[2]],{2,3}]; MRot = Insert[MRot,e31[[3]],{3,3}]; MRot = Insert[MRot,e31[[4]],{4,3}]; MRot = insert[MRot,e41[[1]],{1,4}]; MRot = insert[MRot,e41[[2]],{2,4}]; MRot = insert[MRot,e41[[3]],{3,4}]; MRot = insert[MRot,e41[[4]],{4,4}]; (* Se sutituye (p0^2+p1^2+p2^2+p3^2) por np2 *) MRot = MRot//.{p0^2+p1^2+p2^2+p3^2->np2}; (* Se imprime la Matriz de la Tranformación Rotación *) MatrixForm[MRot] Out[33]//MatrixForm= 1 0 0 0 0 np2 np2 np2 2 p1 p2 + 2 p0 p3 p0 - p1 + p2 - p3 -2 p0 p1 + 2 p2 p3 0 ---------np2 np2 np2 $\begin{array}{c} 2 & p0 & p1 + 2 & p2 & p3 \\ \hline p0 & -p1 & -p2 & +p3 \\ \hline p0 & -p1$ -2 p0 p2 + 2 p1 p3 0 np2 np2 np2 In[34]:= (* Se calcula el determinante de la matriz (simplificando)*) Determinante = Det[MRot]//Simplify Out[35]= 3 np2 In[36]:= (* Se sustituye (p0^2+p1^2+p2^2+p3^2)^3 por np2^3 en el resultado anterior *)

Determinante//.{(p0^2+p1^2+p2^2+p3^2)^3->np2^3}

Out[37]=

1

Se observa que efectivamente se cumple que $Det[M_{p(p,v)}] = +1$.

2.2.1. Significado de los parámetros de la Transformación Rotación p(p,•).

En esta sección se demostrará cual es el significado de los parámetros de la Transformación Rotación ($\rho(p, \bullet)$) analizada hasta ahora. Para ello tenemos lo siguiente:

Sean $\theta \in [0, n\pi]$, $n \in \Re$, $\underline{w} \in \Re^3$, $\underline{w} = 1$ Siendo $p \in Q$ tal que $\rho(p, \underline{w}) = \underline{w}$

Tenemos también que el espacio característico Q_{ϵ} de valor característico λ = 1 se define por, [1] :

$$Q_{\epsilon} = \{ q \in Q : \rho(p,q) = q \}$$
 (2.14)

y el complemento ortogonal de Q_c por:

$$\mathbf{Q}_{\varepsilon}^{\perp} = \{ \mathbf{p} \in \mathbf{Q} : \langle \mathbf{p}, \mathbf{q} \rangle = \mathbf{0}, \forall \mathbf{q} \in \mathbf{Q}_{\varepsilon} \}$$
(2.15)

En base a lo anterior podemos obtener los resultados siguientes.

Teorema 2.2.3: Sea u = $\{u_0, u_1, u_2, u_3\} \in Q_{\epsilon}^{\perp}$, |u| = 1, entonces:

$$\cos \theta = \frac{\langle \mathbf{u}, \rho(\mathbf{p}, \mathbf{u}) \rangle}{\|\mathbf{u}\|^2} = \frac{1}{\|\mathbf{p}\|^2} \langle \mathbf{u}, \mathbf{p} \cdot \mathbf{u} \cdot \overline{\mathbf{p}} \rangle = \frac{1}{\|\mathbf{p}\|^2} (\mathbf{p}_0^2 + \|\mathbf{p}_v\|^2 (2\mathbf{u}_0^2 - 1)) \quad (2.16)$$

$$\operatorname{Sen}^{2} \theta = (\pm) \frac{4 \|\mathbf{p}_{v}\|^{2}}{\|\mathbf{p}\|^{4}} (1 - u_{0}^{2}) (\mathbf{p}_{0}^{2} + u_{0}^{2} \|\mathbf{p}_{v}\|^{2})$$
(2.17)

Demostración: En efecto,

<pre>in(1):= (* Definición de la Transformación Rotación *) (* Se define la Transformación de un Quaternión a un vector tridimensional Tv[] *) Tv[q_]:=(q[[2]],q[[3]],q[[4]]); (* Se define la Transformación inversa de Tv[] *) Tvinv[v_]:=(0.v[[1]],v[[2]],v[[3]]); (* Se define la Transformación de un Quaternión a Quaternión vectorial *) TvQ[q_]:=(0.q[[2]],q[[3]],q[[4]]); (* Se define la Transformación de un Quaternión a Quaternión vectorial *) TvQ[q_]:=(0.q[[2]],q[[3]],q[[4]]); (* Se define la Multiplicación entre dos Quaterniones *) ProductoQ[p]:=(p[(1]]'q[1] + p[2])*q[[2]) - p[[3]]*q[[3]] - p[[4]]*q[[3]],</pre>		
<pre>(*Se define la Transformación de un Quaternión a un vector tridimensional Tv[] *) Tv[q_]=(q[[2]],q[[3]],q[[4]]); (*Se define la Transformación inversa de Tv[] *) Tv[nv[v_]:=(0.v[[1]],v[[2]]),v[[3]]); (*Se define la Transformación de un Quaternión a Quaternión vectorial *) TvQ[q_]=(0.q[[2]),q[[3]],q[[4]]); (*Se define la Multiplicación entre dos Quaterniones *) ProductoQ[p_,q_]=(p[[1]],q[[1]]) + p[[2]]*q[[2]] + p[[3]]*q[[1]] + p[[4]]*q[[4]], p[[1]]*q[[2]] + p[[2]]*q[[2]] + p[[3]]*q[[1]] + p[[4]]*q[[1]], p[[1]]*q[[2]] + p[[2]]*q[[2]] + p[[3]]*q[[1]] + p[[4]]*q[[1]]); (*Se define el conjugado de un Quaternión *) ConjugadoQ[p_]:=(p[[1]],-p[[2]],-p[[3]],-p[[4]]); (*Se define la Transformación Rotación de un vector tridimensional, considerando norma unitaria de *p*, si se desea aplicar otra norma debera incluirse el producto 1/np*2 *) Rotacion[p_,v_]:= Tv[ProductoQ[p,ProductoQ[TvInv[v],ConjugadoQ[p]]]; (*Se define la Transformación Reflexión de un Quaternión, también con norma unitaria de *p* *) RotacionQ[p_,q_]:= RotacionQ[p_,q_]:= ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]; <i>in[16]:=</i> (*Se ilmpian variables *) ClearAll["p*"."u*"."n*".T] (*Se definen los Quaterniones p y u *) p=(p0,p1,p2,p3); u=(u0,u1,u2,u3); (*Se definen los Quaterniones p y u *) p=(p0,p1,p2,p3); u=(u0,u1,u2,u3); (*Se determina el *Coseno de Theta* = coseno *) coseno = u.RotacionQ[p,u]/Expand; (*Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> -2*p1*2*u1*u2 - 2*p2*2*u2*2 + -2*p3*2*u3*2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;</pre>	In[1	/):= (* Definición de la Transformación Rotación *)
<pre>(* Se define la Transformación inversa de Tv[] *) Tvinv[v_]:={0.v[[1]],v[[2]],v[[3]]); (* Se define la Transformación de un Quaternión a Quaternión vectorial *) TvQ[q_]={0.q[[2]],q[[3]],q[[4]]); (* Se define la Mutiplicación entre dos Quaterniones *) ProductoQ[p_q_]={(0,q[[2]),q[[1]] + p[[2]]*q[[1]] + p[[3]]*q[[3]] - p[[4]]*q[[4]], p[[1]]*q[[3]] + p[[2]]*q[[1]] + p[[3]]*q[[1]] + p[[4]]*q[[3]], p[[1]]*q[[3]] + p[[2]]*q[[1]] + p[[3]]*q[[1]] + p[[4]]*q[[1]]); p[[1]]*q[[3]] + p[[2]]*q[[1]] + p[[3]]*q[[1]] + p[[4]]*q[[1]]); (* Se define el conjugado de un Quaternión *) ConjugadoQ[p_]:={p[[1]],-p[[3]],-p[[3]],-p[[4]]); (* Se define la Transformación Rotación de un vector tridimensional, considerando norma unitaria de *p*, si se desea aplicar otra norma debera incluirse el producto 1/np*2 *) Rotacion[p_,v_]:= Tv[ProductoQ[p,ProductoQ[TvInv[v].ConjugadoQ[p]]]; (* Se define la Transformación Reflexión de un Quaternión, también con norma unitaria de *p**) RotacionQ[p_,q_]:= RotacionQ[p_,q_]:= ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]; <i>in[16]:=</i> (* Se limpian variables *) ClearAll["p*","u*","n*",T] (* Se definen los Quaterniones p y u *) p=(p0,p1,p2,p3); u=(u0,u1,u2,u3); (* Se determina el *Coseno de Theta* = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno/,4*p1*p2*u1*u2-> -2*p1^2*u1*2 - 2*p2*2*u2*2 + -2*p3*2*u3*2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;</pre>		(* Se define la Transformación de un Quaternión a un vector tridimensional Tv[] *) Tv[q_]:={q[[2]],q[[3]],q[[4]]};
<pre>(* Se define la Transformación de un Quaternión a Quaternión vectorial *) TvQ[q_]:={0,q[[2]],q[[3]],q[[4]]}: (* Se define la Multiplicación entre dos Quaterniones *) ProductoQ[p]:={p[[1]]*q[[1]] + p[[2]]*q[[2]] - p[[3]]*q[[3]] - p[[4]]*q[[3]], p[[1]]*q[[3]] - p[[2]]*q[[2]] + p[[3]]*q[[4]] + p[[3]]*q[[4]] - p[[4]]*q[[2]], p[[1]]*q[[4]] + p[[2]]*q[[4]] + p[[3]]*q[[4]] + p[[4]]*q[[2]], p[[1]]*q[[4]] + p[[2]]*q[[4]] + p[[3]]*q[[4]] + p[[4]]*q[[1]]; (* Se define el conjugado de un Quaternión *) ConjugadoQ[p_]:=(p[[1]]p[[2]]p[[4]]); (* Se define la Transformación Rotación de un vector tridimensional, considerando norma unitaria de "p", si se dessea aplicar otra norma debera incluirse el producto 1/np"2 *) Rotacion[p_,v_]:= Tv[ProductoQ[p,ProductoQ[TvInv[v].ConjugadoQ[p]]]; (* Se define la Transformación Reflexión de un Quaternión, también con norma unitaria de "p" *) RotacionQ[p_,q_]:= RotacionQ[p_,q_]:= ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]; <i>In[16]:=</i> (* Se limpian variables *) ClearAll["p*"."u*"."n*".T] (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> 2*p1*2*u1*2 - 2*p2*2*u2*2 + - 2*p3*2*u3*2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;</pre>		(* Se define la Transformación inversa de Tv[] *) Tvlnv[v_]:={0,v[[1]],v[[2]],v[[3]]};
<pre>(* Se define la Multiplicación entre dos Quaterniones *) ProductoQ[pq_]:={p[[1]]*q[[1]] * p[[2]]*q[[2]] * p[[3]]*q[[3]] * p[[4]]*q[[3]], p[[1]]*q[[2]] * p[[2]]*q[[4]] * p[[3]]*q[[4]] * p[[4]]*q[[3]], p[[1]]*q[[4]] * p[[2]]*q[[4]] * p[[3]]*q[[4]] * p[[4]]*q[[3]], p[[1]]*q[[4]] * p[[2]]*q[[4]] * p[[3]]*q[[4]] * p[[4]]*q[[3]], p[[1]]*q[[4]] * p[[2]]*q[[4]] * p[[3]]*q[[4]] * p[[4]]*q[[4]] * p[[4]]*q[[4]]); (* Se define el conjugado de un Quaternión *) ConjugadoQ[p_]:={(p[[1]]*p[[2]]*p[[3]]*p[[4]])*; (* Se define el a Transformación Rotación de un vector tridimensional, considerando norma unitaria de "p", si se desea aplicar otra norma debera incluirse el producto 1/np"2 *) Rotacion[p_,v_]:= Tv[ProductoQ[p.ProductoQ[TvInv[v],ConjugadoQ[p]]]); (* Se define la Transformación Reflexión de un Quaternión, también con norma unitaria de "p" *) RotacionQ[p_,q_]:= RotacionQ[p_,q_]:= ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]; <i>In[16]:=</i> (* Se limpian variables *) ClearAll["p*","u*","n*",T] (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1*2*u1*u2 - 2*p2*2*u2*2 +</pre>		(* Se define la Transformación de un Quaternión a Quaternión vectorial *) TvQ[q_]:={0.q[[2]].q[[3]].q[[4]]};
<pre>(* Se define el conjugado de un Quaternión *) ConjugadoQ[p_]:={p[[1]],-p[[2]],-p[[3]],-p[[4]]}; (* Se define la Transformación Rotación de un vector tridimensional, considerando norma unitaria de "p", si se desea aplicar otra norma debera incluirse el producto 1/np"2 *) Rotacion[p_,v_]:= Tv[ProductoQ[p,ProductoQ[TvInv[v],ConjugadoQ[p]]]); (* Se define la Transformación Reflexión de un Quaternión, también con norma unitaria de "p" *) RotacionQ[p_,q_]:= RotacionQ[p_,q_]:= ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]; <i>In[16]:=</i> (* Se limpian variables *) ClearAll["p*","u*","n*",T] (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es lgual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> .2*p1^2*u1^2 - 2*p2^2*u2^2 + .2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;</pre>		(* Se define la Multiplicación entre dos Quaterniones *) ProductoQ[p_,q_]:={p[(1]]*q[[1]] - p[[2]]*q[[2]] - p[[3]]*q[[3]] - p[[4]]*q[[4]], p[[1]]*q[[2]] + p[[2]]*q[[1]] + p[[3]]*q[[4]] - p[[4]]*q[[3]], p[[1]]*q[[3]] - p[[2]]*q[[4]] + p[[3]]*q[[1]] + p[[4]]*q[[2]], p[[1]]*q[[4]] + p[[2]]*q[[3]] - p[[3]]*q[[2]] + p[[4]]*q[[1]]);
<pre>(* Se define la Transformación Rotación de un vector tridimensional, considerando norma unitaria de "p", si se desea aplicar otra norma debera incluirse el producto 1/np"2 *) Rotacion[p_,v_]:= Tv[ProductoQ[p,ProductoQ[TvInv[v],ConjugadoQ[p]]]]; (* Se define la Transformación Reflexión de un Quaternión, también con norma unitaria de "p" *) RotacionQ[p_,q_]:= RotacionQ[p_,q_]:= ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]; <i>in[16]:=</i> (* Se limpian variables *) ClearAll["p*","u*","n*",T] (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]/Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + - 2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;</pre>		(* Se define el conjugado de un Quaternión *) ConjugadoQ[p_]:={p[[1]],-p[[2]],-p[[3]],-p[[4]]);
Rotacion[p_,v_]:= Tv[ProductoQ[p,ProductoQ[TvInv[v],ConjugadoQ[p]]]]; (* Se define la Transformación Reflexión de un Quaternión, también con norma unitaria de "p" *) RotacionQ[p_,q_]:= RotacionQ[p_,q_]:= ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]; In[16]:= (* Se limpian variables *) ClearAll["p**,"u**","n**",T] (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + -2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;		(* Se define la Transformación Rotación de un vector tridimensional, considerando norma unitaria de "p", si se desea aplicar otra norma debera incluirse el producto 1/np^2 *)
<pre>(* Se define la Transformación Reflexión de un Quaternión, también con norma unitaria de "p" *) RotacionQ[p_,q_]:= RotacionQ[p_,q_]:= ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]: In[16]:= (* Se limpian variables *) ClearAll["p*","u*","n*",T] (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 +</pre>		Rotacion[p_,v_]:= Tv[ProductoQ[p,ProductoQ[TvInv[v],ConjugadoQ[p]]]];
RotacionQ[p_,q_]:= RotacionQ[p_,q_]:= ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]: In[16]:= (* Se limpian variables *) ClearAll["p*", "u*", "n*", T] (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + -2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;		(* Se define la Transformación Reflexión de un Quaternión, también con norma unitaria de "p" *)
<pre>////in[16]:= (* Se limpian variables *) ClearAll["p*","u*","n*",T] (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es Igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> -2*p1^2*u1^2 - 2*p2^2*u2^2 +</pre>		RotacionQ[p_,q_]:= RotacionQ[p_,q_]:= ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]];
<pre>(* Se limpian variables *) ClearAll["p*", "u*", "n*", T] (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + - 2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;</pre>	in[1	6]:=
ClearAll["p*", "u*", "n*", T] (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es lgual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + - 2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;		(* Se limpian variables *)
 (* Se definen los Quaterniones p y u *) p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es Igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + -2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3; 		ClearAll["p*", "u*", "n*", T]
<pre>p={p0,p1,p2,p3}; u={u0,u1,u2,u3}; (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + - 2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;</pre>		(* Se definen los Quaterniones p y u *)
 (* Se determina el "Coseno de Theta" = coseno *) coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + - 2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3; 		p={p0,p1,p2,p3}; u={u0,u1,u2,u3};
coseno = u.RotacionQ[p,u]//Expand; (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + - 2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;		(* Se determina el "Coseno de Theta" = coseno *)
(* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *) coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + - 2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;		coseno = u.RotacionQ[p,u]//Expand;
coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + - 2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;		(* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes vectoriales es igual a cero *)
		coseno = coseno//.4*p1*p2*u1*u2-> - 2*p1^2*u1^2 - 2*p2^2*u2^2 + - 2*p3^2*u3^2 - 4*p1*p3*u1*u3 - 4*p2*p3*u2*u3;



Se puede observar que las salidas Out[29] y Out[38] son iguales a lo que se quería demostrar en este teorema.

Teorema 2.2.4: Sea u = (0, u1, u2, u3) $\in Q_\epsilon^-$, entonces:

$$\mathbf{u} \times \mathbf{R}(\mathbf{p}, \mathbf{u}) = \frac{2\mathbf{p}_0 \|\mathbf{u}\|^2}{\|\mathbf{p}\|^2} (\mathbf{0}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) \neq 2\mathbf{p}_0 \frac{\|\mathbf{u}\|^2}{\|\mathbf{p}\|^2} \mathbf{p}_v \qquad (2.18)$$

Demostración: En efecto,

In[1]:=
(* Se define la transformación Tv[], que transforma un Quaternión en un vector tridimensional *)
Tv[q_]:={q[[2]],q[[3]],q[[4]]};
(* Se define el Producto Cruz entre dos vectores tridimensionales *)
Cruz[a_,b_]:= {a[[2]] b[[3]] - a[[3]] b[[2]], - a[[1]] b[[3]] + a[[3]] b[[1]], a[[1]] b[[2]] - a[[2]] b[[1]]}
[n]4] ·=
(* Se definen los Quaterniones u y p *)
u={0,u1,u2,u3}; u={0,u1,u2,u3};
(* Se realiza el producto Cruz de la partes vectoriales de u y R(p,u), incluyendose la norma de "p", que no es unitaria (1/nP^2) *)
uXRu = Cruz[Tv[u],(1/nP^2)*Tv[RotacionQ[p,u]]]//Expand;
(* Se aplica la definición de norma de u y la propiedad de que el producto interno de u y p es igual a cero por ser ortogonales *)
uXRu = uXRu//.{u3^2->nu^2-u1^2-u2^2,p1*u1->-p3*u3-p2*u2}//Expand; uXRu = uXRu//.{u3^2->nu^2-u1^2-u2^2,p3*u3->-p1*u1-p2*u2}//Expand; uXRu = uXRu//.{p2*u2->-p1*u1-p3*u3}//Expand; uXRu = uXRu//.{u1^2->nu^2-u3^2-u2^2}//Expand
Out[12]=
2 nu p0 p1 2 nu p0 p2 2 nu p0 p3
() 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
nr nr nr

El resultado obtenido en Mathematica[®] se puede representar también como:

$$\left(\frac{2\|\mathbf{u}\|^{2}\mathbf{p}_{0}\mathbf{p}_{1}}{\|\mathbf{p}\|^{2}}, \frac{2\|\mathbf{u}\|^{2}\mathbf{p}_{0}\mathbf{p}_{2}}{\|\mathbf{p}\|^{2}}, \frac{2\|\mathbf{u}\|^{2}\mathbf{p}_{0}\mathbf{p}_{3}}{\|\mathbf{p}\|^{2}}\right) = \frac{2\|\mathbf{u}\|^{2}\mathbf{p}_{0}}{\|\mathbf{p}\|^{2}}(\mathbf{p}_{1}, \mathbf{p}_{2}, \mathbf{p}_{3}) = \frac{2\|\mathbf{u}\|^{2}\mathbf{p}_{0}}{\|\mathbf{p}\|^{2}}\mathbf{p}_{v} = \mathbf{u} \times \mathbf{R}(\mathbf{p}, \mathbf{u})$$

Con lo anterior se demuestra la validez de este teorema.

Teorema 2.2.5: Sea $u = \{0, u_1, u_2, u_3\} \in Q_1, u = 1$, entonces:

$$\mathbf{Cos}\,\theta = \frac{1}{\|\mathbf{p}\|^2} (\mathbf{p}_0^2 - \|\mathbf{p}_1\|^2) ; \qquad \mathbf{Sen}\,\theta = (\pm) \frac{2 \|\mathbf{p}_1\|}{\|\mathbf{p}\|^2} \mathbf{p}_0 \qquad (2.19)$$

$$\mathbf{u} \times \rho(\mathbf{p}, \mathbf{u}) = (\pm) \operatorname{Sen} \theta \ \underline{w}$$
(2.20)

Donde $\underline{w} = \frac{1}{\|\mathbf{p}_v\|}\mathbf{p}_v$.

Demostración: En efecto, si tomamos del Teorema 2.2.3 Cos θ , Sen θ y sustituimos $u_0 = 0$, obtenemos directamente lo mostrado para estas dos variables en el teorema. Para el producto cruz $\mathbf{u} \times p(\mathbf{p}, \mathbf{u})$, del Teorema 2.2.4 tenemos que:

$$\mathbf{u} \times \rho(\mathbf{p}, \mathbf{u}) = \frac{2\mathbf{p}_0 \|\mathbf{u}\|^2}{\|\mathbf{p}\|^2} (\mathbf{0}, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = 2\mathbf{p}_0 \frac{\|\mathbf{u}\|^2}{\|\mathbf{p}\|^2} \mathbf{p}_v$$

de este teorema: Sen $\theta = (\pm) \frac{2 \|\mathbf{p}_v\|}{\|\mathbf{p}\|^2} \mathbf{p}_0 \implies \mathbf{p}_0 = (\pm) \frac{\|\mathbf{p}\|^2}{2 \|\mathbf{p}_v\|} \operatorname{Sen} \theta$

Sustituyendo $p_0 y | u | = 1 en la expresión del producto <math>u \times \rho(p, u)$:

$$\mathbf{u} \times \rho(\mathbf{p}, \mathbf{u}) = 2((\pm) \frac{\|\mathbf{p}\|^2}{2\|\mathbf{p}_v\|} \operatorname{Sen} \theta) \frac{\|\mathbf{u}\|^2}{\|\mathbf{p}\|^2} \mathbf{p}_v = (\pm) \frac{\mathbf{p}_v}{\|\mathbf{p}_v\|} \operatorname{Sen} \theta$$

Como $\mathbf{u} \times p(\mathbf{p}, \mathbf{u})$ da como resultado un vector perpendicular al plano formado por $\mathbf{u} \neq p(\mathbf{p}, \mathbf{u})$, que es la dirección del eje de rotación, podemos definir como eje de rotación al vector $\underline{\mathbf{w}} = \frac{\mathbf{p}_v}{\|\mathbf{p}_v\|}$, por lo que:

$$\mathbf{u} \times \rho(\mathbf{p}, \mathbf{u}) = (\pm) \mathbf{Sen} \boldsymbol{\theta} \mathbf{w}$$

_	_	_		

Teorema 2.2.6: Sea $\theta \in [0, n\pi]$, $\mathbf{n} \in \Re$, $\underline{\mathbf{w}} \in \mathbf{Q}_i$, $\|\underline{\mathbf{w}}\| = 1$, entonces la parte vectorial del Quaternión $\mathbf{p} = \{\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\} \in \mathbf{Q}$ con $\mathbf{p}_0 \in \Re$ - {0} es:

$$p_v = \{p_1, p_2, p_3\} = (\pm) \frac{|p|^2}{2p_0} \operatorname{Sen} \theta \underline{w};$$
 (2.21)

siendo <u>w</u> tal que: $\rho(\mathbf{p}, \underline{w}) = \underline{w}$ También $\mathbf{p}_0 \in \Re$ - (0) satisface la siguiente ecuación:

$$4 p_0^4 - 4 |p||^2 p_0^2 \cos \theta - |p||^4 \operatorname{Sen}^2 \theta = 0 \qquad (2.22)$$

Demostración: Del Teorema 2.2.4 se tiene que:

$$\mathbf{u} \times \rho(\mathbf{p}, \mathbf{u}) = (\pm) \operatorname{Sen} \theta \underline{w} = \frac{2\mathbf{p}_0}{\|\mathbf{p}\|^2} \mathbf{p}_v$$

Despejando p, tenemos que:

$$(\pm)\operatorname{Sen} \theta \underline{w} = \frac{2p_0}{\|\mathbf{p}\|^2} \mathbf{p}_{\mathbf{v}} \implies \mathbf{p}_{\mathbf{v}} = (\pm) \frac{\|\mathbf{p}\|^2}{2p_0} \operatorname{Sen} \theta \underline{w}$$

con esto se demuestra la primera parte del teorema. Para la segunda parte tenemos del Teorema 2.2.5 que:

$$\cos \theta = \frac{1}{\|\mathbf{p}\|^2} (\mathbf{p}_0^2 - \|\mathbf{p}_v\|^2) ; \qquad \text{Sen } \theta = (\pm) \frac{2 \|\mathbf{p}_v\|}{\|\mathbf{p}\|^2} \mathbf{p}_0$$

Despejando de Sen θ a $\|\mathbf{p}_{v}\|$ y de Cos θ a \mathbf{p}_{0}^{2} , tenemos que:

$$|\mathbf{p}_{v}|| = (\pm) ||\mathbf{p}||^{2} \frac{\operatorname{Sen} \theta}{2 \, \mathbf{p}_{0}} ; \qquad \mathbf{p}_{0}^{2} = ||\mathbf{p}||^{2} \operatorname{Cos} \theta + ||\mathbf{p}_{v}||^{2}$$

combinando ambos resultados:

$$p_0^2 = \|p\|^2 \cos \theta + ((\pm) \|p\|^2 \frac{\sin \theta}{2p_0})^2 = \|p\|^2 \cos \theta + \|p\|^4 \frac{\sin^2 \theta}{4p_0^2}$$
$$p_0^2 - \|p\|^2 \cos \theta - \|p\|^4 \frac{\sin^2 \theta}{4p_0^2} = 0$$
$$4 p_0^4 - 4 \|p\|^2 p_0^2 \cos \theta - \|p\|^4 \sin^2 \theta = 0$$

Con esto se demuestra la segunda parte del teorema.

Teorema 2.2.7: Las raíces reales de la ecuación de cuarto grado son:

ANTECEDENTES

$$4 p_0^4 - 4 ||p||^2 p_0^2 \cos \theta - ||p||^4 \operatorname{Sen}^2 \theta = 0 \qquad (2.23)$$

$$p_{0_1} = \|p\| \cos \frac{\theta}{2}, \quad p_{0_2} = -\|p\| \cos \frac{\theta}{2}$$
 (2.24)

Demostración: En efecto,

In[1]:=
 (* Se resuelve la ecuación de cuarto grado y se simplifica la solución *)
 Solve[4*p0^4 - 4*nP^2*p0^2*Cos[T] - nP^4*(Sin[T])^2 == 0.{p0}]//Simplify//PowerExpand
 //Simplify//PowerExpand
Out[2]=
 {{p0 -> -I nP Sin[-]}, {p0 -> I nP Sin[-]},
 2
 {p0 -> -(nP Cos[-])}, {p0 -> nP Cos[-]},
 2

Se puede observar que las soluciones reales de la ecuación de cuarto grado, son las presentadas en el teorema.

Finalmente obtenemos con una combinación de los teoremas anteriores:

$$p_0 = (\pm) ||p|| \cos \frac{\theta}{2}$$
 (2.25)

$$\mathbf{p}_{v} = \langle \mathbf{p}_{1}, \mathbf{p}_{2}, \mathbf{p}_{3} \rangle = (\pm) \frac{\|\mathbf{p}\|^{2}}{2\|\mathbf{p}\| \cos \frac{\theta}{2}} \operatorname{Sen} \theta \underline{w} = (\pm) \|\mathbf{p}\| \sin \frac{\theta}{2} \underline{w} \qquad (2.26)$$

Propledades adicionales del álgebra de Quaterniones y de la representación paramétrica de rotaciones se encuentran en [1].[2].

2.3. Representación paramétrica de reflexiones.

De la misma forma que en la sección anterior, aquí presentaremos la definición y propiedades de una transformación lineal que representará paramétricamente una reflexión de cuerpo rígido. Entendiéndose como reflexión a una transformación lineal, ortogonal de determinante negativo. Lo

anterior se mostrará detalladamente mas adelante, cuando se presente el significado de los parámetros de la reflexión.

De la misma forma que para las rotaciones, tenemos la transformación lineal $R(p,\bullet)$: $Q \rightarrow Q$, con p $\in Q$ fijo, como:

$$\mathbf{R}(\mathbf{p},\mathbf{q}) = \overline{\mathbf{p}^* \mathbf{q}^* \mathbf{p}^{-1}} = \frac{1}{\left\|\mathbf{p}\right\|^2} \cdot \left(\overline{\mathbf{p}^* \mathbf{q}^* \mathbf{p}}\right), \quad \forall \mathbf{q} \in \mathbf{Q}, \quad (2.27)$$

Podemos observar que se trata del conjugado de la transformación *Rotación*. De esta transformación se desprenden los siguientes teoremas.

Teorema 2.3.1: La transformación R(p,•): $Q \rightarrow Q$ es lineal, ortogonal y R(p,q) $\in Q_V$, $\forall q \in Q_V$.

Demostración: Para demostrar al linealidad de la transformación, utilizamos las propiedades de asociatividad y distributividad de la estructura algebraica de Q.

$$R(p,(q \oplus s)) = \frac{1}{\|p\|^2} \circ \left\{ \overline{p^*(q \oplus s)^* \overline{p}} \right\} = \frac{1}{\|p\|^2} \circ \left\{ \overline{p^*(q^* \overline{p} \oplus s^* \overline{p})} \right\} = \frac{1}{\|p\|^2} \circ \left\{ \overline{p^*q^* \overline{p} \oplus p^*s^* \overline{p}} \right\} = \frac{1}{\|p\|^2} \circ \left\{ \overline{p^*q^* \overline{p} \oplus p^*s^* \overline{p}} \right\} = R(p,q) \oplus R(p,s), \quad \forall p,q,s \in Q$$

У

$$\mathbf{R}(\mathbf{p}, (\alpha \bullet \mathbf{q})) = \frac{1}{\|\mathbf{p}\|^2} \bullet \left\{ \overline{\mathbf{p}^* (\alpha \bullet \mathbf{q})^* \overline{\mathbf{p}}} \right\} = \frac{\alpha}{\|\mathbf{p}\|^2} \left\{ \overline{\mathbf{p}^* \mathbf{q}^* \overline{\mathbf{p}}} \right\} = \alpha \bullet \mathbf{R}(\mathbf{p}, \mathbf{q}), \quad \forall \mathbf{p}, \mathbf{q} \in \mathbf{Q}, \quad \alpha \in \Re$$

Con lo anterior se demuestra la linealidad de la transformación, es decir que $R(p,\bullet) \in L(Q, Q)$. Ahora para la demostración de la ortogonalidad de $R(p,\bullet)$ utilizamos el Teorema 2.1.2 y el Teorema 2.1.3. Ahora para que la transformación sea ortogonal debe cumplirse que:

$$\langle \mathbf{R}(\mathbf{p},\mathbf{q}), \mathbf{R}(\mathbf{p},\mathbf{s}) \rangle = \langle \mathbf{p},\mathbf{s} \rangle \quad \forall \mathbf{q},\mathbf{s} \in \mathbf{Q}.$$

$$\begin{split} \langle \mathsf{R}(\mathsf{p},\mathsf{q}),\mathsf{R}(\mathsf{p},\mathsf{s})\rangle &= \frac{1}{2\|\mathsf{p}\|^4} \, T_{\mathsf{R}}\left\{ (\overline{\mathsf{p}^* \mathsf{q}^* \overline{\mathsf{p}}})^* (\overline{\mathsf{p}^* \mathsf{s}^* \overline{\mathsf{p}}}) \oplus (\overline{\mathsf{p}^* \mathsf{s}^* \overline{\mathsf{p}}})^* (\overline{\mathsf{p}^* \mathsf{q}^* \overline{\mathsf{p}}}) \right\} \\ &= \frac{1}{2\|\mathsf{p}\|^4} \, T_{\mathsf{R}}\left\{ (\mathsf{p}^* \mathsf{q}^* \overline{\mathsf{p}})^* (\overline{\mathsf{p}^* \mathsf{s}^* \overline{\mathsf{p}}}) \oplus (\mathsf{p}^* \mathsf{s}^* \overline{\mathsf{p}})^* (\overline{\mathsf{p}^* \mathsf{q}^* \overline{\mathsf{p}}}) \right\} \\ &= \frac{1}{2\|\mathsf{p}\|^4} \, T_{\mathsf{R}}\left\{ (\mathsf{p}^* \mathsf{q}^* \overline{\mathsf{p}})^* (\mathsf{p}^* \overline{\mathsf{s}}^* \overline{\mathsf{p}}) \oplus (\mathsf{p}^* \mathsf{s}^* \overline{\mathsf{p}})^* (\mathsf{p}^* \overline{\mathsf{q}^* \overline{\mathsf{p}}}) \right\} \\ &= \frac{1}{2\|\mathsf{p}\|^2} \, T_{\mathsf{R}}\left\{ (\mathsf{p}^* \mathsf{q}^* \overline{\mathsf{s}}^* \overline{\mathsf{p}}) \oplus (\mathsf{p}^* \mathsf{s}^* \overline{\mathsf{q}}^* \overline{\mathsf{p}}) \right\} = \frac{1}{2\|\mathsf{p}\|^2} \, T_{\mathsf{R}}\left\{ (\mathsf{p}^* (\mathsf{q}^* \overline{\mathsf{s}} \oplus \mathsf{s}^* \overline{\mathsf{q}})^* \overline{\mathsf{p}}) \right\} \\ &= \frac{1}{2\|\mathsf{p}\|^2} \, T_{\mathsf{R}}\left\{ (\mathsf{p}^* 2\langle \mathsf{q},\mathsf{s}\rangle^* \overline{\mathsf{p}}) \right\} = \frac{2\langle \mathsf{q},\mathsf{s}\rangle}{2\|\mathsf{p}\|^2} \, T_{\mathsf{R}}\left\{ (\mathsf{p}^* \mathsf{p}) \right\} = \langle \mathsf{q},\mathsf{s}\rangle \end{split}$$

Finalmente observemos que si q = $\{0, q_1, q_2, q_3\} \in Q_V$, entonces $R(p,q) \in Q_V$. En efecto

In[1]:=

(* Se define la transformación reflexión *)

ReflexionQ[p_,q_]:= (1/np2)*ProductoQ[p,ProductoQ[ConjugadoQ[q],ConjugadoQ[p]]];

(* np2= norma de p al cuadrado *)

In[3]:=

(* Se definen Quaterniones p y q *)

p={p0,p1,p2,p3}; q={0,q1,q2,q3};

(* Se determina R(p,q) *)

Collect[ReflexionQ[p,q],{np2,q1,q2,q3}]//InputForm

Out[6]=

Teorema 2.3.2: La transformación R(p,•) es una reflexión.

Demostración: Algunas de las propiedades necesarias para que la transformación sea una reflexión se han demostrado ya en el Teorema 2.2.1, por ello solo resta demostrar que el determinante de la matriz asociada a la transformación $M_{B(p,s)} \in M_{414}$ es -1. En efecto,

```
In[1]:=
     (* Se define el Quaternión p *)
    p = {p0,p1,p2,p3};
    (* se define la base canónica de R4 *)
    e1 = \{1,0,0,0\}; e2 = \{0,1,0,0\}; e3 = \{0,0,1,0\}; e4 = \{0,0,0,1\};
In[5]:=
    (* Se determina la matriz de la transformación (MRef) *)
    (* Se inicializa la lista Mref *)
    MRef={{},{}.{}.{}.{}.{}}:
    (* Se aplica la rotación a la base canónica *)
    e11 = ReflexionQ[p,e1];
    e21 = ReflexionQ[p.e2];
    e31 = ReflexionQ[p.e3];
    e41 = ReflexionQ[p,e4];
    (* Se determinan los elementos de la matriz *)
    MRef = Insert[MRef,e11[[1]],{1,1}];
    MRef = Insert[MRef,e11[[2]].{2,1}];
    MRef = insert[MRef,e11[[3]],{3,1}];
    MRef = insert[MRef,e11[[4]].{4,1}];
    MRef = insert[MRef,e21[[1]],{1,2}];
    MRef = insert[MRef,e21[[2]],{2,2}];
    MRef = Insert[MRef.e21[[3]],{3,2}];
    MRef = insert[MRef,e21[[4]],{4,2}];
    MRef = Insert[MRef,e31[[1]],{1,3}];
    MRef = Insert[MRef,e31[[2]],{2,3}];
    MRef = Insert[MRef,e31[[3]],{3,3}];
    MRef = Insert[MRef,e31[[4]],{4,3}];
    MRef = Insert[MRef,e41[[1]],{1,4}];
    MRef = Insert[MRef,e41[[2]],{2,4}];
    MRef = Insert[MRef,e41[[3]],{3,4}];
    MRef = insert[MRef,e41[[4]],[4,4}];
    (* Se sutituye (p0^2+p1^2+p2^2+p3^2) por np2 *)
    MRef = MRef//.{p0^2+p1^2+p2^2+p3^2->np2};
```

(* Se imprime la Matriz de la Tranformación Reflexion *)

.

MatrixForm[MRef]



Se observa que efectivamente se cumple que $Det[M_{R(p,i)}] = -1$.

2.3.1. Significado de los parámetros de la Transformación Reflexión R(p,•).

De la misma forma que se demostró el significado de los parámetros de la Transformación Rotación $(\rho(p,\bullet))$ en [1], ahora se determinará el significado de los parámetros de la Transformación Reflexión $(R(p,\bullet))$ mostrada anteriormente. Para ello tenemos lo siguiente:

Sean $\theta \in [0, n\pi]$, $n \in \Re$, $\underline{w} \in \Re^3$, $|\underline{w}| = 1$ Siendo $p \in Q$ tal que $R(p, \underline{w}) = -\underline{w}$

Tenemos también que el espacio característico Q_E de valor característico λ = -1 se define por, [1] :

$$Q_{\varepsilon} = \{ q \in Q : R(p,q) = \overline{q} \}$$

$$(2.28)$$

y el complemento ortogonal de Q_c por:

$$\mathbf{Q}_{\varepsilon}^{\perp} = \{ \mathbf{p} \in \mathbf{Q} : \langle \mathbf{p}, \mathbf{q} \rangle = \mathbf{0}, \forall \mathbf{q} \in \mathbf{Q}_{\varepsilon} \}$$
(2.29)

En base a lo anterior podemos obtener los resultados siguientes.

Teorema 2.3.3: Sea $u = \{u_0, u_1, u_2, u_3\} \in Q_6^+$, |u| = 1, entonces:

$$\mathbf{Cos} \ \theta = \frac{\langle \mathbf{u}, \mathbf{R}(\mathbf{p}, \mathbf{u}) \rangle}{\|\mathbf{u}\|^2} = \frac{1}{\|\mathbf{p}\|^2} \langle \mathbf{u}, \ \overline{\mathbf{p}^* \mathbf{u}^* \overline{\mathbf{p}}} \rangle = \frac{1}{\|\mathbf{p}\|^2} \left(\|\mathbf{p}_{\mathbf{v}}\|^2 + \mathbf{p}_0^2 (2u_0^2 - 1) \right)$$
(2.30)

$$\mathbf{Sen}^{2} \theta = (\pm) \frac{4\mathbf{p}_{0}^{2}}{\|\mathbf{p}\|^{4}} (\mathbf{1} - \mathbf{u}_{0}^{2}) (\|\mathbf{p}_{1}\|^{2} + \mathbf{p}_{0}^{2}\mathbf{u}_{0}^{2})$$
(2.31)

Demostración: En efecto,

In[1]:=

(* Definición de la Transformación Reflexión *) (* Se define la Transformación de un Quaternión a un vector Tv[] *) $Tv[q_:={q[[2]],q[[3]],q[[4]]};$ (* Se define la Transformación inversa de Tv[] *) TvInv[v_]:={0,v[[1]],v[[2]],v[[3]]}; (* Se define la Transformación de un Quaternión a Quaternión vectorial *) TvQ[q_]:={0,q[[2]],q[[3]],q[[4]]}; (* Se define la Multiplicación entre dos Quaterniones *) ProductoQ[p_,q_]:={p[[1]]*q[[1]] - p[[2]]*q[[2]] - p[[3]]*q[[3]] - p[[4]]*q[[4]], $p[[1]]^{a}[[2]] + p[[2]]^{a}[[1]] + p[[3]]^{a}[[4]] - p[[4]]^{a}[[3]],$ $p[[1]]^{a}[[3]] - p[[2]]^{a}[[4]] + p[[3]]^{a}[[1]] + p[[4]]^{a}[[2]],$ $p[[1]]^{q}[[4]] + p[[2]]^{q}[[3]] - p[[3]]^{q}[[2]] + p[[4]]^{q}[[1]];$ (* Se define el conjugado de un Quaternión *) ConjugadoQ[p_]:={p[[1]],-p[[2]],-p[[3]],-p[[4]]}; (* Se define la Transformación Reflexión de un vector tridimensional, considerando norma unitaria de "p", si se desea aplicar otra norma debera incluirse el producto 1/np² *) Reflexion[p_,q_]:=Tv[ConjugadoQ[ProductoQ[p.ProductoQ[TvInv[q],ConjugadoQ[p]]]]]];

ANTECEDENTES

```
(* Se define la Transformación Reflexión de un Quaternión, también con norma unitaria
       de "p" *)
    ReflexionQ[p_,q_]:= ConjugadoQ[ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]];
In[16]:=
    (* Se limpian variables *)
    ClearAll["p*","u*","n*",T]
    (* Se definen los Quaterniones p y u *)
    p={p0,p1,p2,p3}; u={u0,u1,u2,u3};
    (* Se determina el Coseno de Theta = coseno *)
    coseno = u.ReflexionQ[p,u]//Expand;
    (* Se aplica la ortogonalidad entre p y u, es decir el producto interno de sus partes
       vectoriales es igual a cero *)
    coseno = coseno//.-4*p1*p2*u1*u2-> 2*p1^2*u1^2 + 2*p2^2*u2^2 +
                      2*p3^2*u3^2 + 4*p1*p3*u1*u3 + 4*p2*p3*u2*u3;
    (* Se usa la norma unitaria de u *)
    coseno = Collect[coseno//.{u3^2->1-u0^2-u1^2-u2^2},{u0}];
    (* Se sustituye la norma al cuadrado de pv (parte vectorial del Quaternión p) por la
       variable nPv^2*)
    coseno = coseno//.{p1^2+p2^2+p3^2->nPv^2};
    (* Como no se trata de norma unitaria de p, se incluye el cuadrado de la norma de p como
       nP^2 *)
    coseno = Collect[coseno,{p0}]/nP^2
Out[29]=
     2 2 2 2
nPv + p0 (-1 + 2 u0)
                    2
                  nP
In[30]:=
    (* Ahora se determina el "seno cuadrado de Theta" = seno2 utilizando la identidad de
       \cos^{2}(T) + Sen^{2}(T) = 1^{*}
     seno2 = 1-coseno^2//Expand;
     (* Sustituimos nP^2 por nP2 y npv^2 por npv2 *)
```

 $seno2 = seno2//.{nP->nP2^(1/2),nPv->nPv2^(1/2)}//Expand;$

```
(* Se aplica la norma de "p", nP^2 = p0^2 + nPv^2 *)
```

seno2 = (seno2//.{nPv2->nP2-p0^2}//Expand//Factor)*nP2^2; seno2 = (seno2//.nP2->nPv2+p0^2//Expand//Factor)/nP2^2;

(* Sustituimos nP2 por nP^2 y npv por npv^2 *)

seno2 = seno2//.{nP2->nP^2,nPv2->nPv^2}

Out[38]=

Se puede observar que las salidas Out[29] y Out[38] son iguales a lo que se quería demostrar en este teorema.

Teorema 2.3.4: Sea u = $\{0, u_1, u_2, u_3\} \in Q_E^{\perp}$, entonces:

$$\mathbf{u} \times \mathbf{R}(\mathbf{p}, \mathbf{u}) = \frac{-2\mathbf{p}_0 \|\mathbf{u}\|^2}{\|\mathbf{p}\|^2} (0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = -2\mathbf{p}_0 \frac{\|\mathbf{u}\|^2}{\|\mathbf{p}\|^2} \mathbf{p}_v \qquad (2.32)$$

Demostración: En efecto,

```
in[1]:=
```

(* Se define la transformación Tv[], que transforma un Quaternión en un vector tridimensional *)

Tv[q_]:={q[[2]],q[[3]],q[[4]]};

(* Se define el Producto Cruz entre dos vectores tridimensionales *)

Cruz(a_,b_):= (a[[2]] b[[3]] - a[[3]] b[[2]], - a[[1]] b[[3]] + a[[3]] b[[1]], a[[1]] b[[2]] - a[[2]] b[[1]])

in[4]:=

(* Se definen los Quaterniones u y p *)

u={0,u1,u2,u3}; u={0,u1,u2,u3};

(* Se realiza el producto Cruz de la partes vectoriales de u y R(p,u), incluyendose la norma de "p", que no es unitaria (1/nP^2) *)

uXRu = Cruz[Tv[u],(1/nP^2)*Tv[ReflexionQ[p,u]]]//Expand;

(* Se aplica la definición de norma de u y la propiedad de que el producto interno de u y p es igual a cero por ser ortogonales *) uXRu = uXRu//.{u3^2->nu^2-u1^2-u2^2.p1*u1->-p3*u3-p2*u2}//Expand; uXRu = uXRu//.{u3^2->nu^2-u1^2-u2^2 p3*u3->-p1*u1-p2*u2}//Expand: uXRu = uXRu//.{p2*u2->-p1*u1-p3*u3}//Expand; uXRu = uXRu//.{u1^2->nu^2-u3^2-u2^2}//Expand Out[12]= 2 2 -2 nu p0 p1 -2 nu p0 p2 -2 nu p0 p3 2 2 2 nP nP nP

El resultado obtenido en Mathematica® se puede representar también como:

$$\left(\frac{-2\|\mathbf{u}\|^{2}\mathbf{p}_{0}\mathbf{p}_{1}}{\|\mathbf{p}\|^{2}}, \frac{-2\|\mathbf{u}\|^{2}\mathbf{p}_{0}\mathbf{p}_{2}}{\|\mathbf{p}\|^{2}}, \frac{-2\|\mathbf{u}\|^{2}\mathbf{p}_{0}\mathbf{p}_{3}}{\|\mathbf{p}\|^{2}}\right) = \frac{-2\|\mathbf{u}\|^{2}\mathbf{p}_{0}}{\|\mathbf{p}\|^{2}}(\mathbf{p}_{1}, \mathbf{p}_{2}, \mathbf{p}_{3}) = \frac{-2\|\mathbf{u}\|^{2}\mathbf{p}_{0}}{\|\mathbf{p}\|^{2}}\mathbf{p}_{1} = \mathbf{u} \times \mathbf{R}(\mathbf{p}, \mathbf{u})$$

Con lo anterior se demuestra la validez de este teorema.

Teorema 2.3.5: Sea u = {0, u1, u2, u3} ∈ Q, u = 1, entonces:

$$\cos \theta = \frac{1}{\|\mathbf{p}\|^2} (\|\mathbf{p}_v\|^2 - \mathbf{p}_0^2) ; \qquad \operatorname{Sen} \theta = (\pm) \frac{2 \|\mathbf{p}_v\|}{\|\mathbf{p}\|^2} \mathbf{p}_0 \qquad (2.33)$$

$$\mathbf{u} \times \mathbf{R}(\mathbf{p}, \mathbf{u}) = (\pm) \operatorname{Sen} \theta \, \underline{\mathbf{w}} \tag{2.34}$$

Donde $\underline{w} = \frac{1}{\|\mathbf{p}_v\|}\mathbf{p}_v$.

Demostración: En efecto, si tomamos del Teorema 2.3.3 Cos θ , Sen θ y sustituimos $u_0 = 0$, obtenemos directamente lo mostrado para estas dos variables en el teorema. Para el producto cruz $u \times R(p,u)$, del Teorema 2.3.4 tenemos que:

$$\mathbf{u} \times \mathbf{R}(\mathbf{p}, \mathbf{u}) = \frac{-2\mathbf{p}_0 \|\mathbf{u}\|^2}{\|\mathbf{p}\|^2} (0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3) = -2\mathbf{p}_0 \frac{\|\mathbf{u}\|^2}{\|\mathbf{p}\|^2} \mathbf{p}_v$$

de este teorema: Sen $\theta = (\pm) \frac{2 \|\mathbf{p}_v\|}{\|\mathbf{p}\|^2} \mathbf{p}_0 \implies \mathbf{p}_0 = (\pm) \frac{\|\mathbf{p}\|^2}{2 \|\mathbf{p}_v\|} \operatorname{Sen} \theta$

30

Sustituyendo po y u = 1 en la expresión del producto u×R(p,u):

$$u \times R(p,u) = -2((\pm)\frac{\|p\|^2}{2\|p_v\|} \text{Sen } \theta)\frac{\|u\|^2}{\|p\|^2}p_v = (\pm)\frac{\|p_v\|}{\|p_v\|} \text{Sen } \theta$$

Como $\mathbf{u} \times \mathbf{R}(\mathbf{p}, \mathbf{u})$ da como resultado un vector perpendicular al plano formado por $\mathbf{u} \times \mathbf{R}(\mathbf{p}, \mathbf{u})$, que es la dirección del eje de rotación, podemos definir como eje de rotación el vector $\underline{\mathbf{w}} = \frac{\mathbf{p}_v}{\|\mathbf{p}_v\|}$, por lo que:

$$\mathbf{u} \times \mathbf{R}(\mathbf{p}, \mathbf{u}) = (\pm) \mathbf{Sen} \boldsymbol{\theta} \mathbf{w}$$

Teorema 2.3.6: Sea $\theta \in [0, n\pi]$, $n \in \Re$, $\underline{w} \in Q_v$, $\|\underline{w}\| = 1$, entonces la parte vectorial del Quaternión $p = \{p_0, p_1, p_2, p_3\} \in Q$ con $p_0 \in \Re$ - (0) es:

$$\mathbf{p}_{v} = \{\mathbf{p}_{1}, \mathbf{p}_{2}, \mathbf{p}_{3}\} = (\pm) \frac{\|\mathbf{p}\|^{2}}{2\mathbf{p}_{0}} \operatorname{Sen} \theta \underline{w};$$
 (2.35)

siendo w tal que: R(p, w) = - w

También po ∈ ℜ- (0) satisface la siguiente ecuación:

$$4 p_0^4 + 4 \|p\|^2 p_0^2 \cos \theta - \|p\|^4 \operatorname{Sen}^2 \theta = 0$$
 (2.36)

Demostración: Del Teorema 2.3.4 se tiene que:

$$\mathbf{u} \times \mathbf{R}(\mathbf{p}, \mathbf{u}) = (\pm) \operatorname{Sen} \theta \underline{\mathbf{w}} = \frac{-2\mathbf{p}_0}{\|\mathbf{p}\|^2} \mathbf{p}_v$$

Despejando p. tenemos que:

$$(\pm) \operatorname{Sen} \theta \underline{w} = \frac{-2p_0}{\|p\|^2} p_v \implies p_v = (\pm) \frac{\|p\|^2}{2p_0} \operatorname{Sen} \theta \underline{w}$$

con esto se demuestra la primera parte del teorema. Para la segunda parte tenemos del Teorema 2.3.5 que:

$$\cos \theta = \frac{1}{\|\mathbf{p}\|^{2}} (\|\mathbf{p}_{v}\|^{2} - \mathbf{p}_{0}^{2}); \qquad \text{Sen } \theta = (\pm) \frac{2\|\mathbf{p}_{v}\|}{\|\mathbf{p}\|^{2}} \mathbf{p}_{0}$$

Despejando de Sen θ a $\|\mathbf{p}_v\|$ y de Cos θ a \mathbf{p}_0^2 , tenemos que:
$$|\mathbf{p}_{v}|| = (\pm) ||\mathbf{p}||^{2} \frac{\operatorname{Sen} \theta}{2 \mathbf{p}_{0}} ; \qquad \mathbf{p}_{0}^{2} = - ||\mathbf{p}||^{2} \operatorname{Cos} \theta + ||\mathbf{p}_{v}||^{2}$$

combinando ambos resultados:

$$p_{0}^{2} = -\|p\|^{2} \cos \theta + ((\pm) \|p\|^{2} \frac{\sin \theta}{2p_{0}})^{2} = -\|p\|^{2} \cos \theta + \|p\|^{4} \frac{\sin^{2} \theta}{4p_{0}^{2}}$$

$$p_{0}^{2} + \|p\|^{2} \cos \theta - \|p\|^{4} \frac{\sin^{2} \theta}{4p_{0}^{2}} = 0$$

$$4 p_{0}^{4} + 4 \|p\|^{2} p_{0}^{2} \cos \theta - \|p\|^{4} \sin^{2} \theta = 0$$

Con esto se demuestra la segunda parte del teorema.

Teorema 2.3.7: Las raíces reales de la ecuación de cuarto grado son:

$$4 p_0^4 + 4 \left\| p \right\|^2 p_0^2 \cos \theta - \left\| p \right\|^4 \operatorname{Sen}^2 \theta = 0 \qquad (2.37)$$

$$p_{0_1} = \|p\| \operatorname{Sen} \frac{\theta}{2}, \quad p_{0_2} = -\|p\| \operatorname{Sen} \frac{\theta}{2}$$
 (2.38)

Demostración: En efecto,

In[1]:=

(* Se resuelve la ecuación de cuarto grado y se simplifica la solución *)

Solve[4*p0^4 + 4*nP^2*p0^2*Cos[T] - nP^4*(Sin[T])^2 == 0,{p0}]//Simplify//PowerExpand //Simplify//PowerExpand

Out[2]=

{ {
$$p0 \rightarrow -I \ nP \ Cos[-]$$
 }, { $p0 \rightarrow I \ nP \ Cos[-]$ },
 { $p0 \rightarrow -I \ nP \ Sin[-]$ },
 { $p0 \rightarrow -(nP \ Sin[-])$ }, { $p0 \rightarrow nP \ Sin[-]$ } }

.

Se puede observar que las soluciones reales de la ecuación de cuarto grado, son las presentadas en el teorema.

Finalmente obtenemos con una combinación de los teoremas anteriores:

$$p_0 = (\pm) \|p\| \operatorname{Sen} \frac{\theta}{2}$$
 (2.39)

$$\mathbf{p}_{\mathbf{v}} = \{\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3\} = (\pm) \frac{\|\mathbf{p}\|^2}{2|\mathbf{p}|| \operatorname{Sen} \frac{\theta}{2}} \operatorname{Sen} \theta \underline{w} = (\pm) \|\mathbf{p}\|| \operatorname{Cos} \frac{\theta}{2} \underline{w} \qquad (2.40)$$

.

-

CAPITULO III

3. MODELACION CINEMATICA

En este capítulo se presentará la modelación de la cinemática de algunos manipuladores robóticos utilizando el álgebra de Quaterniones. Dicha modelación se realizará utilizando inicialmente solamente rotaciones, después se obtendrán otros modelos diferentes aplicando reflexiones o una combinación de rotaciones y reflexiones según sea el caso. Lo anterior nos permitirá tener dos modelos cinemáticos diferentes para el mismo manipulador, que nos permitirá mover un manipulador, siguiendo una misma trayectoria, pero con una configuración diferente de sus eslabones en el espacio. Esto puede aplicarse, por ejemplo, en el proceso de evasión de obstáculos, ya que el manipulador puede cambiar la configuración de sus eslabones permitiéndole evadir algunos obstáculos al cambiar de configuración sin afectar la trayectoria.

3.1. MODELACION CINEMATICA DE UN MANIPULADOR DE TRES GRADOS DE LIBERTAD, UTILIZANDO ROTACIONES CON DIFERENTES SECUENCIAS DE MOVIMIENTOS.

OBJETIVO: El principal objetivo de esta sección, será presentar las diferencias en el proceso de modelación de la cinemática de un manipulador de tres grados de libertad, cuando se realiza utilizando diferentes secuencias de movimientos de sus eslabones. Se mostrará además, que el modelo final es el mismo, pero con configuraciones intermedias diferentes. Dicho cuerpo se presenta en la Figura 3.1

3.1.1. Configuración no deformada.

Mostraremos primeramente la modelación de la configuración no deformada, que consiste en determinar un sistema de ecuaciones que nos permitan determinar la posición del segundo y tercer eslabón, cuando el primer eslabón se encuentra en una posición inicial de referencia (home), conociendo la posición del elemento terminal del manipulador. Para lo anterior tenemos dos posibles secuencias de movimientos:

Primera Secuencia:

1.- Se mantiene el primer eslabón fijo. Rotación 2.- Se aplica una rotación al segundo eslabón. (Figura 3.2). Rotación 3.- Se aplica una rotación al tercer eslabón. (Figura 3.3).

Segunda Secuencia:

Se mantiene el primer eslabón fijo.
 Rotación 2.- Se aplica una rotación al tercer eslabón. (Figura 3.4).
 Rotación 3.- Se aplica una rotación al segundo eslabón. (Figura 3.5).

Para lo anterior definiremos la configuración no deformada del manipulador, que consiste en definir el vector de posición del elemento terminal para posiciones arbitrarias del segundo y tercer eslabones en un plano dado. Consideremos la base canónica de \Re^3 , esto es:

 $\mathbf{e}_1 = \{1,0,0\}; \quad \mathbf{e}_2 = \{0,1,0\}; \quad \mathbf{e}_3 = \{0,0,1\};$

Ubicamos sistemas de referencia locales en la base de cada eslabón del manipulador como se muestra en la Figura 3.1. Observemos que, en este caso, los sistemas de referencia definidos coinciden con la base canónica la cual sirve como sistema de referencia inerclal fijo, es decir:

$$\mathbf{g}_{i} = \mathbf{f}_{i} = \mathbf{e}_{i}$$
, $j = 1, 2, 3$.



Figura 3.1. Bases locales en la configuración original para la modelación.

De la posición mostrada en la Figura 3.1, debemos permitir que los eslabones 2 y 3 giren a una posición arbitraria, considerando que los eslabones tienen los siguientes ejes de rotación:

Eslabón	Eje De Rotación
1	e,
2	$f_1 = e_1$
3	$\mathbf{g}_1 = \mathbf{e}_1$

Con los ejes de rotación anteriores, podemos definir la forma que tendrán inicialmente los Quaterniones que nos permitirán rotar los eslabones 2 y 3.

Eslabón 2: Consideremos el siguiente parámetro que caracterizará a nuestra primera rotación;

 $P = \{P_0, P_1, P_2, P_3\} \implies P_v = \{P_1, P_2, P_3\}, ||P|| = 1.$

En este caso se tiene que siendo $\underline{w} \in \Re^3$ el eje de rotación, entonces, [1]:

$$\begin{split} & \mathsf{P}_{\mathsf{o}} = \mathsf{Cos}(\varphi_1/2) \\ & \mathsf{P}_{\mathsf{v}} = \mathsf{Sin}(\varphi_1/2)^* \underline{\mathsf{w}} = \mathsf{Sin}(\varphi_1/2)^* \mathbf{f}_1 = \mathsf{Sin}(\varphi_1/2)^* \mathbf{e}_1 = \mathsf{Sin}(\varphi_1/2)^* \{1, 0, 0\} \\ & \mathsf{P}_{\mathsf{v}} = \{\mathsf{P}_1, 0, 0\} \end{split}$$

De lo anterior se obtiene la forma local del parámetro P, que es: P = {P₀, P₁, 0, 0}, Donde:

 $P_0 = Cos(\phi_1/2);$ $P_1 = Sen(\phi_1/2);$

Siendo ϕ_1 el ángulo de rotación del eslabón 2.

Eslabón 3: Consideremos ahora el siguiente parámetro que caracterizará la segunda rotación;

 $Q = \{Q_0, Q_1, Q_2, Q_3\} \Rightarrow Q_v = \{Q_1, Q_2, Q_3\}, ||Q|| = 1.$

Siendo $w \in \Re^3$ el eje de rotación, tenemos que:

 $\begin{aligned} \mathbf{Q}_{v} &= \mathbf{Sin}(\phi_{2}/2)^{*} \, \underline{\mathbf{w}} \ = \mathbf{Sin}(\phi_{2}/2)^{*} \mathbf{g}_{1} = \mathbf{Sin}(\phi_{2}/2)^{*} \mathbf{e}_{1} = \mathbf{Sin}(\phi_{2}/2)^{*} \{1,0,0\} \\ \mathbf{Q}_{v} &= \{ \mathbf{Q}_{1}, 0, 0 \} \end{aligned}$

De lo anterior se obtiene la forma local del parámetro Q, que es: Q = {Q₀, Q₁, 0, 0}, Donde:

$$Q_0 = Cos(\phi_2/2);$$

 $Q_1 = Sen(\phi_2/2);$

Siendo φ_2 el ángulo de rotación del eslabón 3.

Con los Quaterniones anteriores podemos ahora realizar las rotaciones necesarias para determinar la configuración no deformada, se debe observar que si los ejes de rotación cambian su forma, también cambiará la forma de los Quaterniones dados.

Aplicación de la Primera Secuencia en la configuración no deformada.

Rotación 2.- Primeramente rotamos el segundo eslabón con el Quaternión P. Esto se realiza rotando la base local f_j , pero además, debemos rotar la base g_j ya que al rotar el eslabón 2, el eslabón tres también se ve afectado por la misma rotación (Figura 3.2).

Como la rotación anterior afectó el eje de rotación del tercer eslabón (g_1), dicho eje de rotación cambia, siendo el nuevo eje g_1^t . Por lo tanto el Quaternión **Q** se modifica de la siguiente forma:

 $\begin{aligned} \mathbf{Q}^{1} &= (\mathbf{Q}_{0}^{1}, \mathbf{Q}_{1}^{1}, \mathbf{Q}_{2}^{1}, \mathbf{Q}_{3}^{1}), \text{ con } \mathbf{Q}_{0}^{1} &= \operatorname{Cos}(\varphi_{2}/2), \qquad \mathbf{Q}_{v}^{1} &= \operatorname{Sin}(\varphi_{2}/2)^{*} \mathbf{g}_{1}^{1} \\ \mathbf{Q}_{v}^{1} &= \operatorname{Sin}(\varphi_{2}/2)^{*} \rho(\mathbf{P}, \mathbf{g}_{1}), \end{aligned}$ $\begin{aligned} \operatorname{Como} \operatorname{Sin}(\varphi_{2}/2) &= \operatorname{Cte}, \text{ se tiene que: } \mathbf{Q}_{v}^{1} &= \rho(\mathbf{P}, \mathbf{Q}_{v}) \implies \mathbf{Q}^{1} &= \rho(\mathbf{P}, \mathbf{Q}) \end{aligned}$

El Quaternión Q¹ caracteriza a la rotación de la barra 3. Esto es, el Quaternión es actualizado, con respecto al sistema inercial fijo, puesto que el eje original de rotación es afectado por las correspondientes rotaciones locales de los eslabones, lo cual modifica con respecto al sistema de

referencia global al mismo. Si aplicamos la transformación $\rho(P,Q)$ obtenemos la forma explicita del Quaternión Q^1 :

 $\mathbf{Q}^{i} = \rho(\mathbf{P}, \mathbf{Q}) = \{\mathbf{Q}_{0}, \mathbf{P}_{0}^{*}(-\mathbf{P}_{1}^{*}\mathbf{Q}_{0} + \mathbf{P}_{0}^{*}\mathbf{Q}_{1}) + \mathbf{P}_{1}^{*}(\mathbf{P}_{0}^{*}\mathbf{Q}_{0} + \mathbf{P}_{1}^{*}\mathbf{Q}_{1}), 0, 0\}$

 $Q^1 = \{Q_0, (P_0^2 + P_1^2), Q_1, 0, 0\} = \{Q_0, Q_1, 0, 0\} = Q$

Podemos observar, en este caso, que el Quaternión Q no sufrió ninguna transformación, esto debido a que los ejes de rotación de los eslabones 2 y 3 son paralelos.



Figura 3.2. Rotación de los eslabones 2 y 3 en la primera secuencia, para la configuración no deformada.

Rotación 3.- De la misma forma, ahora se rotará el tercer eslabón utilizando el Quaternión correspondiente Q^i , lo que implica que la base g_j sufre otra transformación como se ilustra a continuación. (Figura 3.3)

 $\mathbf{g}_i^{II} = \rho(\mathbf{Q}^I, \mathbf{g}_i^I)$ (Rotación del tercer eslabón con Quaternión \mathbf{Q}^I).

Pero como $\mathbf{g}_{i}^{1} = \rho(\mathbf{P}, \mathbf{g}_{i}), y \mathbf{Q}^{1} = \rho(\mathbf{P}, \mathbf{Q})$ se tiene que: $\mathbf{g}_{i}^{11} = \rho(\rho(\mathbf{P}, \mathbf{Q}), \rho(\mathbf{P}, \mathbf{g}_{i}))$

Desarrollando las rotaciones anteriores tenemos que:

 $\mathbf{g}_{i}^{H} = \rho(\mathbf{P}^{*}\mathbf{Q}^{*} \mathbf{\overline{P}}_{i} \mathbf{P}^{*} \mathbf{g}_{i}^{*} \mathbf{\overline{P}}) = (\mathbf{P}^{*}\mathbf{Q}^{*} \mathbf{\overline{P}}^{*} \mathbf{P}^{*} \mathbf{g}_{i}^{*} \mathbf{\overline{P}}^{*} \mathbf{\overline{P}}^{*} \mathbf{Q}^{*} \mathbf{\overline{P}})$

Pero como: $\overline{P^*Q^*P} = P^*\overline{P^*Q}$ y $\overline{P}^*P = P^*\overline{P} = \{1,0,0,0\}$, entonces:

$$\mathbf{g}_{j}^{II} = \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{g}_{j}) = \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{g}_{j})) = \mathbf{P}^{*}\mathbf{Q}^{*} \mathbf{g}_{j}^{*} \mathbf{\overline{P}^{*}\mathbf{Q}}$$
(3.2)

MODELACION CINEMATICA



Figura 3.3. Rotación del eslabón 3 en la primera secuencia, para la configuración no deformada.

Aplicación de la Segunda Secuencia en la configuración no deformada.

Rotación 1.- Rotamos el tercer eslabón con el Quaternión Q. Esto se realiza rotando solamente la base local g_j, es decir, esta rotación no afecta a ningún otro eslabón y por ello tampoco afecta a ninguna otra base local ni Quaternión. (Figura 3.4)

 $g_i^{l} = \rho(Q, g_i)$ (Rotación del tercer eslabón con Quaternión Q), j = 1,2,3





Rotación 2.- Se aplica la rotación al segundo eslabón con el Quaternión P, lo que implica que las bases f_i y g_i sufren las siguientes transformaciones.(Figura 3.5)

$$\mathbf{f}_{i}^{\dagger} = \rho(\mathbf{P}, \mathbf{f}_{i})$$
 (Rotación del segundo eslabón con Quaternión P) (3.3)

 $\mathbf{g}_{i}^{II} = \rho(\mathbf{P}, \mathbf{g}_{i}^{I})$ (Rotación del tercer eslabón con Quaternión P).

Como $\mathbf{g}_{i}^{l} = \rho(\mathbf{Q}, \mathbf{g}_{i})$ tenemos que:

$$g_{j}^{II} = \rho(P, \rho(Q, g_{j})) = \rho((P^{*}Q), g_{j})$$
 (3.4)

Nota: En este caso no hubo necesidad de modificar el Quaternión Q, debido a que cuando se aplicó la rotación P, que es la que afecta a Q, la rotación correspondiente al Quaternión Q ya se habla aplicado.





De lo anterior podemos observar que la forma de las bases locales, después de aplicar las rotaciones en las dos secuencias, es exactamente la misma, o sea:

La forma explicita de las bases anteriores es la siguiente:

$$f_{1}^{1} = \{1, 0, 0\}$$

$$f_{2}^{1} = \{0, -1 + 2^{\circ}P_{0}^{2}, 2^{\circ}P_{0}^{\circ}P_{1}\} \qquad (3.6)$$

$$f_{3}^{1} = \{0, -2^{\circ}P_{0}^{\circ}P_{1}, -1 + 2^{\circ}P_{0}^{2}\}$$

$$g_{1}^{11} = \{1, 0, 0\}$$

$$g_{2}^{11} = \{0, -P_{0}^{2} + P_{1}^{2} + 2^{\circ}P_{0}^{2*}Q_{0}^{2} - 2^{\circ}P_{1}^{2*}Q_{0}^{2} - 4^{\circ}P_{0}^{\circ}P_{1}^{\circ}Q_{0}^{\circ}Q_{1},$$

$$2^{*}(-\{P_{0}^{*}P_{1}\} + 2^{*}P_{0}^{*}P_{1}^{*}Q_{0}^{2} + P_{0}^{2*}Q_{0}^{*}Q_{1} - P_{1}^{2*}Q_{0}^{*}Q_{1}\})$$

$$g_{3}^{II} = \{0, 2^{*}(P_{0}^{*}P_{1} - 2^{*}P_{0}^{*}P_{1}^{*}Q_{0}^{2} - P_{0}^{2*}Q_{0}^{*}Q_{1} + P_{1}^{2*}Q_{0}^{*}Q_{1}), -P_{0}^{2*} + P_{1}^{2*} + 2^{*}P_{0}^{2*}Q_{0}^{2} - 2^{*}P_{1}^{2*}Q_{0}^{2} - 4^{*}P_{0}^{*}P_{1}^{*}Q_{0}^{*}Q_{1}\}$$

$$(3.7)$$

Podemos observar que la forma final de las bases locales, después de aplicadas las dos secuencias de rotaciones, son idénticas, esto quiere decir que la secuencia en la modelación no afecta la forma final del modelo, pero es importante hacer notar que los puntos intermedios que se generan en una secuencia y otra, son diferentes, como se puede observar en las siguientes figuras.



Figura 3.6. Puntos intermedios en la configuración no deformada.

Los puntos intermedios en las dos diferentes secuencias, quedan determinados por las bases locales correspondientes y por la posición del elemento terminal en cada caso, como se muestra a continuación, siendo L_1 , L_2 y L_3 las longitudes de los estabones.

Vectores que definen los eslabones y posición del elemento terminal, para el punto intermedio de la primera secuencia (Figura 3.6).

Eslabón 1.

 $b_{1} = L_{1}^{*} e_{3} = \{0, 0, L_{1}\};$ Eslabón 2. $b_{2} = L_{2}^{*} f_{3}^{1} = L_{2}^{*} \rho(P, f_{3}) = L_{2}^{*} (P^{*} e_{3}^{*} \overline{P}) = \{0, -2^{*}L_{2}^{*}P_{0}^{*}P_{1}, L_{2}^{*}(2^{*}P_{0}^{2} - 1)\};$ Eslabón 3. $b_{3} = L_{3}^{*} g_{3}^{1} = L_{3}^{*} \rho(P, g_{3}) = L_{3}^{*} (P^{*} e_{3}^{*} \overline{P}) = \{0, -2^{*}L_{3}^{*}P_{0}^{*}P_{1}, L_{3}^{*}(2^{*}P_{0}^{2} - 1)\};$ (3.8)

Posición del elemento terminal en el punto intermedio de la primera secuencia.

$$\mathbf{R}_{p1} = \mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3 = \{\mathbf{0}_1 - \mathbf{2}^*(\mathbf{L}_2 + \mathbf{L}_3)^*\mathbf{P}_0^*\mathbf{P}_1, \mathbf{L}_1 - \mathbf{L}_2 - \mathbf{L}_3 + \mathbf{2}^*(\mathbf{L}_2 + \mathbf{L}_3)^*\mathbf{P}_0^{2}\}$$
(3.9)

Vectores que definen los eslabones y posición del elemento terminal, para el punto intermedio de la segunda secuencia (Figura 3.6).

Eslabón 1.
b₁ = L₁* e₃ = {0, 0, L₁};
Eslabón 2.
b₂ = L₂* f₃ = L₂* e₃ = {0, 0, L₂};
Eslabón 3.
b₃ = L₃* g¹₃ = L₃*
$$\rho(Q, g_3) = L_3* (Q * e_3 * \overline{Q}) = {0, -2*L_3*Q_0*Q_3, L_3*(2*Q_0^2 - 1)};$$

Posición del elemento terminal en el punto intermedio de la segunda secuencia.

Sector Contractor Contractor

$$\mathbf{R}_{p1} = \mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3 = \{\mathbf{0}, -\mathbf{2}^*\mathbf{L}_3^*\mathbf{Q}_0^*\mathbf{Q}_1, \mathbf{L}_1 + \mathbf{L}_2 + \mathbf{L}_3^*(\mathbf{2}^*\mathbf{Q}_0^2 - \mathbf{1})\}$$
(3.11)

3.1.2. Vectores que definen los eslabones y posición del elemento terminal en la configuración no deformada.

De la Figura 3.5 y Figura 3.7 podemos determinar los vectores que definen los eslabones del manipulador y el vector de posición del elemento terminal (punto p). Las longitudes de los eslabones son L1, L2, L3.

b₁ = L₁* e₃; (Eslabón 1)
b₂ = L₂* f₃¹ = L₂*
$$\rho(\mathbf{P}, \mathbf{f}_3) = L_2* (\mathbf{P} * \mathbf{e}_3 * \mathbf{\overline{P}});$$
 (Eslabón 2) (3.12)
b₃ = L₃* g₃¹¹ = L₃* $\rho((\mathbf{P}^*\mathbf{Q}), \mathbf{g}_3) = L_3* ((\mathbf{P}^*\mathbf{Q}) * \mathbf{e}_3* (\mathbf{\overline{P}^*Q}));$ (Eslabón 3)

$$\mathbf{R}_{\mathbf{p}} = \mathbf{b}_{\mathbf{1}} + \mathbf{b}_{\mathbf{2}} + \mathbf{b}_{\mathbf{3}} \quad (\text{Posición del elemento terminal}) \quad (3.13)$$



Figura 3.7. Vector de posición del elemento terminal en la configuración no deformada.

3.1.3. Ecuaciones explícitas optimizadas para la configuración no deformada.

A continuación se muestra la forma de las ecuaciones optimizadas, correspondientes a la configuración no deformada.

Donde:

A1 =
$$P_0^{*}P_1$$
; A2 = 1 - 2* $P_0^{*}2$;
A3 = $Q_0^{*}Q_1$; A4 = 1 - 2* $Q_0^{*}2$; (3.16)

Posición del elemento terminal en el punto intermedio de la segunda secuencia.

$$\mathbf{R}_{p1} = \mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3 = \{\mathbf{0}, -\mathbf{2}^*\mathbf{L}_3^*\mathbf{Q}_0^*\mathbf{Q}_1, \mathbf{L}_1 + \mathbf{L}_2 + \mathbf{L}_3^*(\mathbf{2}^*\mathbf{Q}_0^2 - 1)\}$$
(3.11)

3.1.2. Vectores que definen los eslabones y posición del elemento terminal en la configuración no deformada.

De la Figura 3.5 y Figura 3.7 podemos determinar los vectores que definen los eslabones del manipulador y el vector de posición del elemento terminal (punto p). Las longitudes de los eslabones son L₁, L₂, L₃.

$$\begin{aligned} \mathbf{b}_1 &= \mathbf{L}_1^* \ \mathbf{e}_3; \quad (\text{ Eslabón 1}) \\ \mathbf{b}_2 &= \mathbf{L}_2^* \mathbf{f}_3^1 = \mathbf{L}_2^* \ \rho(\mathbf{P}, \mathbf{f}_3) = \mathbf{L}_2^* (\mathbf{P}^* \mathbf{e}_3^* \ \overline{\mathbf{P}}); \quad (\text{ Eslabón 2}) \\ \mathbf{b}_3 &= \mathbf{L}_3^* \mathbf{g}_3^{11} = \mathbf{L}_3^* \rho((\mathbf{P}^* \mathbf{Q}), \mathbf{g}_3) = \mathbf{L}_3^* ((\mathbf{P}^* \mathbf{Q})^* \mathbf{e}_3^* (\overline{\mathbf{P}^* \mathbf{Q}})); \quad (\text{ Eslabón 3}) \end{aligned}$$

$$\mathbf{R}_{\mathbf{p}} = \mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3$$
 (Posición del elemento terminal) (3.13)



Figure 3.7. Vector de posición del elemento terminal en la configuración no deformada.

3.1.3. Ecuaciones explícitas optimizadas para la configuración no deformada.

A continuación se muestra la forma de las ecuaciones optimizadas, correspondientes a la configuración no deformada.

Donde:

3.1.4. Configuración deformada.

De la misma forma que se realizó la modelación de la configuración no deformada, ahora se modelara la configuración deformada, utilizando dos diferentes secuencias de movimientos de los tres eslabones, partiendo de la configuración deformada. Las secuencias que se utilizarán son las siguientes:

Primera Secuencia.

Rotación 1.- rotan los tres eslabones con el Quaternión r1, partiendo de la configuración no deformada. (Figura 3.8).



Figura 3.8. Rotación de los tres eslabones en la configuración deformada, primera secuencia.

Rotación 2.- rotan los eslabones 2 y 3 con el Quaternión r2. (Figura 3.9).



Figura 3.9. Rotación de los eslabones 2 y 3 en la configuración deformada, primera secuencia.

Rotación 3.- rota el eslabón 3 con el Quaternión r3. (Figura 3.10).





Segunda Secuencia.

Rotación 1.- rota el eslabón 3 con el Quaternión r3, partiendo de la configuración no deformada. (Figura 3.11).



Figura 3.11. Rotación del eslabón 3 en la configuración deformada, segunda secuencia.

Rotación 2.- rotan los eslabones 2 y 3 con el Quaternión r2. (Figura 3.12).



Figura 3.12. Rotación de los eslabones 2 y 3 en la configuración deformada, segunda secuencia.

Rotación 3.- rotan los tres eslabones con el Quaternión rf. (Figura 3.13).



Figure 3.13. Rotación de los tres eslabones en la configuración deformada, segunda secuencia.

De la Figura 3.3 podemos ver que los ejes de rotación de los eslabones se conservan como en la configuración original (Figura 3.1). Por lo anterior podemos definir los Quaterniones para rotar a posición deformada como se muestra a continuación.

Eslabón 1: $r1 = \{r1_0, r1_1, r1_2, r1_3\} \implies r1_v = \{r1_1, r1_2, r1_3\}, ||r1|| = 1.$

La forma local del Quaternión r1 es:

 $r1_{v} = Sin(\theta_{1}/2)^{*} e_{3} = Sin(\theta_{1}/2)^{*}\{0, 0, 1\}$ $r1_{v} = \{0, 0, r1_{3}\}$ De lo anterior: $r1 = \{ r1_0, 0, 0, r1_3 \}$, Donde:

 $r1_0 = Cos(\theta_1/2);$ $r1_3 = Sen(\theta_1/2);$

Siendo 8, el ángulo de rotación del eslabón 1 en la configuración deformada.

Eslabón 2: $r2 = \{r2_0, r2_1, r2_2, r2_3\} \implies r2_v = \{r2_1, r2_2, r2_3\}, ||r2|| = 1.$

La forma local del Quaternión r2 es:

 $r2_v = Sin(\theta_2/2)^* e_1 = Sin(\theta_2/2)^* \{1, 0, 0\}$ $r2_v = \{ r2_1, 0, 0 \}$

De lo anterior: r2 = { r2,, r2,, 0, 0}, Donde:

$$r_{2_0} = Cos(\theta_2/2);$$

 $r_{2_1} = Sen(\theta_2/2);$

Siendo θ₂ el ángulo de rotación del eslabón 2 en la configuración deformada.

Eslabón 3: $r3 = \{r3_0, r3_1, r3_2, r3_3\} \implies r3_v = \{r3_1, r3_2, r3_3\}, ||r3|| = 1.$

La forma local del Quaternión r3 es:

 $r_{3_v} = Sin(\theta_3/2)^* e_1 = Sin(\theta_3/2)^* \{1, 0, 0\}$ $r_{3_v} = \{r_{3_1}, 0, 0\}$

De lo anterior: $r3 = \{ r3_0, r3_1, 0, 0 \}$, Donde:

```
r_{3_0} = Cos(\theta_3/2);
r_{3_1} = Sen(\theta_3/2);
```

Siendo θ_3 el ángulo de rotación del eslabón 3 en la configuración deformada.

Aplicación de la Primera Secuencia en la configuración deformada.

Rotación 1,- Rotación de los tres eslabones con Quaternión r1:

En la Figura 3.8 podemos observar que todas las bases locales son afectadas por la rotación r1 (rotación de los tres eslabones), a continuación se muestran dichas bases.

$$e_j^{I} = \rho(r1, e_j)$$
 (Rotación del primer eslabón con Quaternión r1)
 f_j^{II} = \rho(r1, f_j^{I}) (Rotación del segundo eslabón con Quaternión r1) (3.17)
 g_j^{III} = \rho(r1, g_j^{II}) (Rotación del tercer eslabón con Quaternión r1)
 Donde: j = 1,2,3

Los ejes de rotación del segundo y tercer eslabones son afectados por r1, es decir los nuevos ejes de rotaciones de los eslabones 2 y 3 son f_1^{II} y g_1^{III} respectivamente, por ello los Quaterniones r2 y r3 se transforman de la siguiente forma:

$$r2^{i} = \{r2_{0}, r2^{i}_{v}\}; \quad r2^{i}_{v} = Sin(\theta_{2}/2)^{*} f_{1}^{11} = Sin(\theta_{2}/2)^{*} \rho(r1, f_{1}^{1})$$

$$r2^{i}_{v} = Sin(\theta_{2}/2)^{*} \rho(r1, e_{1}) = \rho(r1, r2_{v}) \implies r2^{i} = \rho(r1, r2)$$

$$(3.18)$$

$$\begin{aligned} \mathbf{r3}^{1} &= \{\mathbf{r3}_{0}, \mathbf{r3}^{1}_{v}\}; \quad \mathbf{r3}^{1}_{v} &= \mathbf{Sin}(\theta_{3}/2)^{*} \rho(\mathbf{r1}, \mathbf{g}_{1}^{11}) \\ \mathbf{r3}^{1}_{v} &= \mathbf{Sin}(\theta_{3}/2)^{*} \rho(\mathbf{r1}, \mathbf{e}_{1}) = \rho(\mathbf{r1}, \mathbf{r3}_{v}) \implies \mathbf{r3}^{1} = \rho(\mathbf{r1}, \mathbf{r3}) \end{aligned}$$

Después de aplicada la primera rotación, los Quaterniones que caracterizan la rotaciones de los eslabones 2 y 3 son r2¹ y r3¹ respectivamente, los cuales han sido actualizados con respecto al sistema inercial fijo, y su forma explícita es la siguiente:

$$\begin{split} \mathbf{r} \mathbf{2}^{1} &= \rho(\mathbf{r}\mathbf{1},\,\mathbf{r}\mathbf{2}) = \{\mathbf{r}\mathbf{2}_{0},\,(-\mathbf{1}\,+\,\mathbf{2}^{*}\mathbf{r}\mathbf{1}_{0}^{2})^{*}\mathbf{r}\mathbf{2}_{1},\,\mathbf{2}^{*}\mathbf{r}\mathbf{1}_{0}^{*}\mathbf{r}\mathbf{1}_{3}^{*}\mathbf{r}\mathbf{2}_{1},\,\mathbf{0}\},\\ \mathbf{r}\mathbf{3}^{1} &= \rho(\mathbf{r}\mathbf{1},\,\mathbf{r}\mathbf{3}) = \{\mathbf{r}\mathbf{3}_{0},\,(-\mathbf{1}\,+\,\mathbf{2}^{*}\mathbf{r}\mathbf{1}_{0}^{2})^{*}\mathbf{r}\mathbf{3}_{1},\,\mathbf{2}^{*}\mathbf{r}\mathbf{1}_{0}^{*}\mathbf{r}\mathbf{1}_{3}^{*}\mathbf{r}\mathbf{3}_{1},\,\mathbf{0}\}. \end{split}$$

Rotación 2.- Rotación de los eslabones 2 y 3 con r21:

Para este caso observamos que las bases locales afectadas por la rotación r2¹ son las que corresponden a los eslabones 2 y 3 (f_i y g_i). (Figura 3.9)

Con la rotación r2¹ se ve afectado el eje de giro del tercer eslabón, que ahora será g_1^{IV} , por ello el Quaternión r3¹ se transforma de la siguiente forma:

$$\begin{aligned} \mathbf{r3^{u}} &= \{\mathbf{r3_{o}}, \mathbf{r3^{u}}_{v}\}; \\ \mathbf{r3^{u}}_{v} &= \mathbf{Sin}(\theta_{3}/2)^{*} \mathbf{g}_{1}^{1V} = \mathbf{Sin}(\theta_{3}/2)^{*} \rho(\mathbf{r2^{l}}, \mathbf{g}_{1}^{111}) = \mathbf{Sin}(\theta_{3}/2)^{*} \rho(\mathbf{r2^{l}}, \rho(\mathbf{r1}, \mathbf{g}_{1}^{11})) \\ \mathbf{r3^{u}}_{v} &= \rho(\mathbf{r2^{l}}, \mathbf{Sin}(\theta_{3}/2)^{*} \rho(\mathbf{r1}, \mathbf{e}_{1})) = \rho(\mathbf{r2^{l}}, \mathbf{r3^{l}}_{v}) \implies \mathbf{r3^{u}} = \rho(\mathbf{r2^{l}}, \mathbf{r3^{l}}) \\ \mathbf{r3^{u}} &= \rho(\mathbf{r2^{l}}, \mathbf{r3^{l}}) = \rho(\rho(\mathbf{r1}, \mathbf{r2}), \rho(\mathbf{r1}, \mathbf{r3})) = \rho(\mathbf{r1^{*}r2^{*}r1}, \mathbf{r1^{*}r3^{*}r1}) \\ \mathbf{r3^{u}} &= \mathbf{r1^{*}r2^{*}r1^{*}r1^{*}r3^{*}r1^{*}r1^{*}r2^{*}r1} \\ \mathbf{r3^{u}} &= \mathbf{r1^{*}r2^{*}r3^{*}r1^{*}r2} = \rho(\mathbf{r1^{*}r2,r3}) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \mathbf{r3})) \end{aligned}$$
(3.20)

Ahora, después de aplicadas las rotaciones anteriores el Quaternión que caracteriza la rotación del tercer eslabón es r3^{II}, cuya forma explícita es:

$$\mathbf{r3}^{\mu} = \rho((\mathbf{r1}^*\mathbf{r2}), \mathbf{r3}) = \{\mathbf{r3}_0, (-1 + 2^*\mathbf{r1}_0^2)^*\mathbf{r3}_1, 2^*\mathbf{r1}_0^*\mathbf{r1}_3^*\mathbf{r3}_1, 0\}.$$

Rotación 3.- Rotación del tercer eslabón 3 con r311:

Por último se tiene la rotación del tercer eslabón con el Quaternión r3", con lo que se modifica la ubicación de la base local del eslabón 3. (Figura 3.10).

$$\mathbf{g}_{j}^{V} = \rho(\mathbf{r3}^{H}, \mathbf{g}_{j}^{IV})$$
 (Rotación del tercer eslabón con Quaternión r3^H) (3.21)
Donde: $\mathbf{j} = 1,2,3$

Finalmente las bases que representan a cada eslabón, con las rotaciones correspondientes son:

Eslabón 1:

$$e_{j} = \rho(r1, e_{j})$$
 (3.22)

Eslabón 2: Simplificando fill.

$$\begin{split} f_{j}^{III} &= \rho(r2^{i}, f_{j}^{II}) = \rho(\rho(r1, r2), \rho(r1, f_{j}^{I})) = \rho(\rho(r1, r2), \rho(r1, \rho(P, f_{j}))) \\ f_{j}^{III} &= \rho(\rho(r1, r2), \rho(r1, \rho(P, f_{j}))) = \rho(r1^{*}r2^{*}r\overline{1}, \rho(r1, P^{*}f_{j}^{*}\overline{P})) \\ f_{j}^{III} &= \rho(r1^{*}r2^{*}r\overline{1}, r1^{*}P^{*}f_{j}^{*}\overline{P}^{*}r\overline{1}) = r1^{*}r2^{*}r\overline{1}^{*}r1^{*}P^{*}f_{j}^{*}\overline{P}^{*}r\overline{1}^{*}r\overline{1}^{*}r\overline{2}^{*}\overline{r1} \\ f_{j}^{III} &= r1^{*}r2^{*}P^{*}f_{j}^{*}\overline{P}^{*}r\overline{1}^{*}r1^{*}r\overline{1}^{*}r\overline{2} = r1^{*}r2^{*}P^{*}f_{j}^{*}\overline{P}^{*}r\overline{1}^{*}r\overline{2} \end{split}$$

.

f ||| = r1*r2*P*f, *r1*r2*P $\mathbf{f}_{j}^{111} = \rho(\mathbf{r1^{*}r2^{*}P}, \mathbf{f}_{j}) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{P}, \mathbf{f}_{j})))$

$$\mathbf{f}_{i}^{111} = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{e}_{i}) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{P}, \mathbf{e}_{i})))$$
(3.23)

Eslabón 3: Simplificando g

$$\begin{aligned} \mathbf{g}_{i}^{V} &= \rho(\mathbf{r3^{u}}, \mathbf{g}_{j}^{IV}) \\ \mathbf{r3^{u}} &= \mathbf{r1^{*}r2^{*}r3^{*}} \overline{\mathbf{r1^{*}r2}} = \mathbf{r1^{*}r2^{*}r3^{*}} \overline{\mathbf{r2^{*}r1}} \\ \mathbf{g}_{i}^{IV} &= \rho(\mathbf{r2^{i}}, \mathbf{g}_{j}^{IU}) = \rho(\rho(\mathbf{r1}, \mathbf{r2}), \rho(\mathbf{r1}, \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{g}_{j})))) \\ \mathbf{g}_{i}^{IV} &= \rho(\mathbf{r1^{*}r2^{*}r1}, \rho(\mathbf{r1}, \rho(\mathbf{P}, \mathbf{Q^{*}g_{j}^{*}}\mathbf{\overline{Q}}))) = \rho(\mathbf{r1^{*}r2^{*}r1}, \rho(\mathbf{r1}, \mathbf{P^{*}Q^{*}g_{j}^{*}\mathbf{\overline{Q}^{*}}\mathbf{\overline{P}})) \\ \mathbf{g}_{i}^{IV} &= \rho(\mathbf{r1^{*}r2^{*}r1}, \mathbf{r1^{*}P^{*}Q^{*}g_{j}^{*}\mathbf{\overline{Q}^{*}}\mathbf{\overline{P^{*}r1}}) \\ \mathbf{g}_{i}^{IV} &= r\mathbf{r1^{*}r2^{*}r1^{*}r1^{*}P^{*}Q^{*}g_{j}^{*}\mathbf{\overline{Q}^{*}}\mathbf{\overline{P^{*}r1^{*}r1^{*}r2^{*}r1} \\ \mathbf{g}_{i}^{IV} &= \mathbf{r1^{*}r2^{*}P^{*}Q^{*}g_{j}^{*}\mathbf{\overline{Q}^{*}}\mathbf{\overline{P^{*}r1^{*}r1^{*}r1^{*}r2^{*}r1} \\ \mathbf{g}_{i}^{IV} &= \mathbf{r1^{*}r2^{*}P^{*}Q^{*}g_{j}^{*}\mathbf{\overline{Q}^{*}}\mathbf{\overline{P}^{*}r2^{*}r1} \\ \mathbf{g}_{i}^{IV} &= \mathbf{r1^{*}r2^{*}P^{*}Q^{*}g_{j}^{*}\mathbf{\overline{Q}^{*}}\mathbf{\overline{P}^{*}r2^{*}r1} \end{aligned}$$

Sustituyendo r3" y $g_j^{\rm IV}$ en $g_j^{\rm V}$, tenemos que:

$$\begin{split} \mathbf{g}_{j}^{V} &= \rho(\mathbf{r}1^{*}\mathbf{r}2^{*}\mathbf{r}3^{*}\mathbf{r}\overline{2}^{*}\mathbf{r}\overline{1}, \mathbf{r}1^{*}\mathbf{r}2^{*}\mathbf{P}^{*}\mathbf{Q}^{*}\mathbf{g}_{j}^{*}\mathbf{\overline{Q}}^{*}\mathbf{\overline{P}}^{*}\mathbf{r}\overline{2}^{*}\mathbf{r}\overline{1}) \\ \mathbf{g}_{j}^{V} &= \mathbf{r}1^{*}\mathbf{r}2^{*}\mathbf{r}3^{*}\mathbf{\overline{r}2}^{*}\mathbf{r}\overline{1}^{*}\mathbf{r}1^{*}\mathbf{r}2^{*}\mathbf{P}^{*}\mathbf{Q}^{*}\mathbf{g}_{j}^{*}\mathbf{\overline{Q}}^{*}\mathbf{\overline{P}}^{*}\mathbf{r}\overline{2}^{*}\mathbf{r}\overline{1}^{*}\mathbf{r}1^{*}\mathbf{r}2^{*}\mathbf{r}3^{*}\mathbf{\overline{r}2}^{*}\mathbf{r}\overline{1} \\ \mathbf{g}_{j}^{V} &= \mathbf{r}1^{*}\mathbf{r}2^{*}\mathbf{r}3^{*}\mathbf{P}^{*}\mathbf{Q}^{*}\mathbf{g}_{j}^{*}\mathbf{\overline{Q}}^{*}\mathbf{\overline{P}}^{*}\mathbf{r}\overline{2}^{*}\mathbf{r}\overline{1}^{*}\mathbf{r}1^{*}\mathbf{r}2^{*}\mathbf{r}\overline{3} \\ \mathbf{g}_{j}^{V} &= \mathbf{r}1^{*}\mathbf{r}2^{*}\mathbf{r}3^{*}\mathbf{P}^{*}\mathbf{Q}^{*}\mathbf{g}_{j}^{*}\mathbf{\overline{r}1^{*}\mathbf{r}2^{*}\mathbf{r}3^{*}\mathbf{P}^{*}\mathbf{Q} \\ \mathbf{g}_{j}^{V} &= \rho(\mathbf{r}1^{*}\mathbf{r}2^{*}\mathbf{r}3^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{g}_{j}) = \rho(\mathbf{r}1, \rho(\mathbf{r}2, \rho(\mathbf{r}3, \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{g}_{j}))))) \end{split}$$

$$\mathbf{g}_{i}^{V} = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{e}_{i}) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{r3}, \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{e}_{i})))))$$
(3.24)

Aplicación de la Segunda Secuencia en la configuración deformada.

Rotación 1.- Rotación del tercer eslabón con r3. Esta rotación solamente afecta al eslabón 3. (Figura 3.11).

$$\mathbf{g}_{j}^{III} = \rho(\mathbf{r3}, \mathbf{g}_{j}^{II})$$
 (Rotación del tercer eslabón con Quaternión r3)
Donde: j = 1,2,3 (3.25)

Rotación 2.- Rotación de los eslabones 2 y 3 con r2. Esta rotación afecta al eje de rotación del tercer eslabón, o sea, al Quaternión r3, el cual ya ha sido aplicado a las bases correspondientes, y por ello no es necesario determinar la nueva forma que adquiere. (Figura 3.12).

Rotación 3.- Rotación de los tres eslabones con r1. Como esta es la última rotación no será necesario actualizar ningún eje de rotación. (Figura 3.13).

$e_j^1 = \rho(r1, e_j)$	(Rotación del primer eslabón con Quaternión r1)	
$\mathbf{f}_{j}^{\mathrm{III}} = \rho(\mathbf{r1}, \ \mathbf{f}_{j}^{\mathrm{II}})$	(Rotación del segundo eslabón con Quaternión r1)	(3.27)
$\mathbf{g}_{j}^{V} = \rho(\mathbf{r1}, \mathbf{g}_{j}^{W})$	(Rotación del tercer eslabón con Quaternión r1)	
Donde: j = 1,2,3		

De la misma forma que en la primera secuencia de movimientos, la forma de las bases locales de cada eslabón es la siguiente:

Eslabón 1.-

$$e_{j}^{l} = \rho(r1, e_{j})$$
 (3.28)

Eslabón 2.-

$$\begin{aligned} \mathbf{f}_{j}^{III} &= \rho(\mathbf{r1}, \mathbf{f}_{j}^{II}) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \mathbf{f}_{j}^{I})) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{P}, \mathbf{f}_{j}))) = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{f}_{j}) \\ \mathbf{f}_{j}^{III} &= \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{P}, \mathbf{e}_{j}))) = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{e}_{j}) \end{aligned}$$
(3.29)

Eslabón 3.-

$$\begin{aligned} \mathbf{g}_{j}^{V} &= \rho(\mathbf{r1}, \mathbf{g}_{j}^{IV}) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \mathbf{g}_{j}^{III})) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{r3}, \mathbf{g}_{j}^{II}))) \\ \mathbf{g}_{j}^{V} &= \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{r3}, \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{g}_{j}))))) = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{g}_{j}) \\ \mathbf{g}_{j}^{V} &= \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{r3}, \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{e}_{j}))))) = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{e}_{j}) \end{aligned}$$
(3.30)

Al igual que en la configuración no deformada, la forma final de las bases locales en las dos secuencias es la misma, cambiando únicamente las posiciones intermedias, las cuales pueden observase en las figuras 3.8 a 3.13. En estas figuras podemos observar que existen dos puntos intermedios en cada secuencia y las ecuaciones que los representan se mostrarán, en su forma no explícita, a continuación:

Puntos intermedios en la Primera Secuencia.

Punto Intermedio 1. (Figura 3.8)

$b_1 = L_1^* e_3^* = L_1^* \rho(r1, e_3);$	(Eslabón 1)	
$b_2 = L_2^* f_3^{11} = L_2^* \rho(r1, f_3^1);$	(Eslabón 2)	(3.31)
$b_3 = L_3^* g_3^{III} = L_3^* \rho(r1, g_3^{II});$	(Eslabón 3)	
$R_{p1} = b_1 + b_2 + b_3;$	(Posición del elemento terminal)	(3.32)

Punto Intermedio 2. (Figura 3.9)

Puntos intermedios en la Segunda Secuencia.

Punto Intermedio 3. (Figura 3.11)

Punto Intermedio 4. (Figura 3.12)

$b_1 = L_1^* e_3;$	(Eslabón 1)	
$b_2 = L_2^* f_3^{11} = L_2^* \rho(r2, f_3^1);$	(Eslabón 2)	(3.37)
$b_3 = L_3^* g_3^{(V)} = L_3^* \rho(r2, g_j^{(H)}) = L_3^* \rho(r2, \rho(r3, g_3^{(H)}))$	(Eslabón 3)	
$\mathbf{R}_{\mathbf{p}} = \mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3;$ (Posición del eleme	ento terminal)	(3.38)

3.1.5. Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada.

De las figuras Figura 3.13 y Figura 3.14 se determinan los vectores que definen los eslabones del manipulador y el vector del posición del elemento terminal (punto p). Las longitudes de los eslabones son L_1 , L_2 , L_3 .

$b_1 = L_1^* e_3^*;$	(Eslabón 1)	
b2 = L2 f3 ;;	(Eslabón 2)	(3.39)
b, = L, g; ;	(Eslabón 3)	

$$r_p = b_1 + b_2 + b_3$$
 (Posición del elemento terminal) (3.40)





3.1.6. Ecuaciones explícitas optimizadas para la configuración deformada.

Las expresiones anteriores se explicitaron y optimizaron utilizando Mathematica, y se presentan a continuación.

$$r_{p} = \{-4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}L_{3}), -2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}L_{3}), L_{1} + B1^{*}L_{2} + C2^{*}L_{3}\}$$
(3.42)

Donde:

A1 = P.*P.; A2 = 1-2"P.2: $A3 = Q_0^*Q_1;$ A4 = 1-2"Q.2; A5 = r1,*r1,; A6 = 1-2*r1,2; A7 = r2, "r2,; A8 = 1-2*r2.2 A9 = r3, r3; A10 = 1-2"r3,2; (3.43)B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8: B3 = A10*A8 - 4*A7*A9; B4 = A10*A7 + A8*A9; B5 = A2*A4 - 4*A1*A3: B6 = A2*A3 + A1*A4: C1 = B4*B5 + B3*B6: C2 = B3*B5 - 4*B4*B6:

De las dos secuencias utilizadas en la modelación mostrada, y habiendo realizado otras secuencias que no se presentan aquí, podemos concluir que:

- Las secuencias de movimientos, en el proceso de modelación, no afecta la forma final de las ecuaciones resultantes, pero se presentan diferentes configuraciones intermedias en cada secuencia.
- 2.- Las secuencias de movimientos pueden facilitar o dificultar la obtención de las ecuaciones finales, ya que en unas secuencias se requiere actualizar ejes de rotación en los Quaterniones correspondientes (*Primera Secuencia*), lo que provoca que las ecuaciones resultantes sean más complejas y deban ser simplificadas utilizando propiedades de las operaciones con Quaterniones, mientras que en otras secuencias de movimientos no se requiere actualizar ejes de rotación (*Segunda Secuencia*), lo que permite que las ecuaciones obtenidas ya tengan su forma final simplificada.
- 3.- La forma más eficiente de modelar la cinemática de cadenas ablertas con Quaterniones es iniciando con el eslabón terminal y finalizar con el eslabón base.
- 4.- La forma general de la rotación de la base local de un eslabón cualquiera en la configuración deformada de una cadena abierta con juntas rotacionales, es:

$$\mathbf{e}^{n}_{j} = \rho(\mathbf{r}_{j}, \mathbf{r}_{2}^{*}, \mathbf{e}^{*}_{n}, \mathbf{e}^{*}_{j})$$
(3.44)

Donde:

- eⁿ, = Base local del eslabón n.
- F₁, F₂, F_n = Quaterniones que rotan los eslabones 1, 2, ... n. Siendo el eslabón base el número uno, y el eslabón n, el eslabón analizado.
- e^x_i = Base local del eslabón n en la configuración no deformada.
- n = Número de eslabón analizado.

La numeración de los eslabones se realiza empezando en el eslabón base, es decir en el que se soporta todo el mecanismo, y se continua secuencialmente hasta finalizar en el eslabón terminal.

3.2. MODELACION CINEMATICA DE UN MANIPULADOR DE TRES GRADOS DE LIBERTAD, UTILIZANDO ROTACIONES Y REFLEXIONES.

OBJETIVO: El objetivo en este caso será determinar un modelo cinemático diferente al obtenido utilizando solamente rotaciones, el cual servirá para ubicar los eslabones del manipulador en una posición alterna en la cinemática inversa, esto con el propósito de evitar un posible obstáculo que pudiese impedir el movimiento del manipulador.

Para realizar la modelación, partiremos de la configuración no deformada utilizada en la modelación con rotaciones (Figura 3.5), aplicaremos una rotación al primer eslabón del manipulador y posteriormente aplicaremos reflexiones a los otros dos eslabones, la actualización de los ejes de rotación se realizará utilizando la transformación correspondiente, solo que tomaremos como eje de rotación el de sentido negativo, que es equivalente a realizar una actualización de eje de rotación solo con reflexiones como se verá más delante; por lo anterior procederemos a obtener directamente la configuración deformada. Debemos observar que la configuración inicial del manipulador es diferente a la que se obtuvo para la modelación con rotaciones (Figura 3.15).



Figura 3.15. Configuración inicial y no deformada del manipulador de 3 grados de libertad.



Figura 3.16. Rotación de los tres eslabones en la configuración deformada.

3.2.1. Configuración deformada.

Para obtener la configuración deformada utilizaremos la siguiente secuencia.

Rotación 1.- rotan los tres eslabones con el Quaternión r1. (Figura 3.16).

Reflexión 1.- se aplican reflexiones a los eslabones 2 y 3 con el Quaternión r2. (Figura 3.17).



Figura 3.17. Reflexión de los eslabones 2 y 3 en la configuración deformada.

Reflexión 2.- se aplica reflexión al eslabón 3 con el Quaternión r3. (Figura 3.18).



Figura 3.18. Reflexión del eslabón 3 en la configuración deformada.

Los Quaterniones que se utilizarán para el giro de los eslabones serán los mismos que los utilizados en el modelo realizado solamente con rotaciones, es decir, la forma local de dichos Quaterniones es:: Eslabón 1: r1 = (r1₀, r1₁, 0, 0), Donde:

 $r1_0 = Cos(\theta_1/2);$ $r1_1 = Sen(\theta_1/2);$

Siendo 0, el ángulo de rotación del eslabón 1 en la configuración deformada.

Eslabón 2: r2 = { r20, 0, 0, r23}.

 $r_{2_0} = Sen(\theta_2/2);$ $r_{2_1} = Cos(\theta_2/2);$

Siendo 82 el ángulo de rotación del eslabón 2 en la configuración deformada.

```
Eslabón 3: r3 = { r30, 0, 0, r33}.
```

 $r_{3_0} = Sen(\theta_3/2);$ $r_{3_1} = Cos(\theta_3/2);$

Siendo 03 el ángulo de rotación del eslabón 3 en la configuración deformada.

Con lo anterior procedemos a la aplicación de las rotaciones y reflexiones mencionadas.

Rotación 1.- Rotación de los tres eslabones con r1:

En la Figura 3.16 podemos observar que todas las bases locales son afectadas por la rotación r1 (rotación de los tres eslabones), a continuación se muestran dichas bases.

$\mathbf{e}_{j}^{1} = \rho(\mathbf{r1}, \mathbf{e}_{j})$	(Rotación del primer eslabón con Quaternión r1)	
$\mathbf{f}_{j}^{\mathrm{H}} = \rho(\mathbf{r}1, \mathbf{f}_{j}^{\mathrm{H}})$	(Rotación del segundo eslabón con Quaternión r1)	(3.45)
$\mathbf{g}_{j}^{\mathrm{III}} = \rho(\mathbf{r1}, \mathbf{g}_{j}^{\mathrm{II}})$	(Rotación del tercer estabón con Quaternión r1)	
Donde: j = 1,2,3		

Los ejes de rotación del segundo y tercer eslabones son afectados por r1, es decir los nuevos ejes de rotación serían f_1^{II} y g_1^{III} según la modelación realizada con rotaciones, pero en este caso se realizará la variante mencionada anteriormente, es decir, los nuevos ejes de rotación serán - f_1^{II} y - g_1^{III} .

$$-f_{1}^{11} = -\rho(\mathbf{r1}, f_{1}^{1})$$
 (Eje de rotación del eslabón 2).
$$-g_{1}^{11} = -\rho(\mathbf{r1}, g_{1}^{11})$$
 (Eje de rotación del eslabón 3). (3.46)

Por lo anterior la forma actualizada de los Quaterniones r2 y r3 es:

$$r2^{i} = \{r2_{0}, r2^{i}_{\nu}\}; \quad r2^{i}_{\nu} = Cos(\theta_{2}/2)^{*} (-f_{1}^{11}) = Cos(\theta_{2}/2)^{*} (-\rho(r1, f_{1}^{11}))$$

$$r2^{i}_{\nu} = -Cos(\theta_{2}/2)^{*} \rho(r1, e_{1}) = \rho(r1, -r2_{\nu})$$

$$\Rightarrow \quad r2^{i} = \overline{\rho(r1, r2)} = R(r1, r2) \quad (3.47)$$

$$r3^{i} = \{r3_{0}, r3^{i}_{v}\}; \quad r3^{i}_{v} = \cos(\theta_{3}/2)^{*} (-g_{1}^{i1}) = \cos(\theta_{3}/2)^{*} (-\rho(r1, g_{1}^{11}))$$

$$r3^{i}_{v} = -\cos(\theta_{3}/2)^{*} \rho(r1, e_{1}) = \rho(r1, -r3_{v})$$

$$\Rightarrow r2^{i} = \overline{\rho(r1, r3)} = R(r1, r3) \quad (3.48)$$

Después de aplicada la primera rotación, los Quaterniones correspondientes a los eslabones 2 y 3 son r2¹ y r3¹ respectivamente, los cuales han sido actualizados con respecto al sistema inercial fijo mediante una Reflexión como se mostró, y su forma explícita es la siguiente:

$$\begin{aligned} \mathbf{r2}^{i} &= \mathbf{R}(\mathbf{r1}, \, \mathbf{r2}) = \{\mathbf{r2}_{0}, \, (\mathbf{1} - \mathbf{2}^{*}\mathbf{r1}_{0}^{2})^{*}\mathbf{r2}_{1}, \, - \, \mathbf{2}^{*}\mathbf{r1}_{0}^{*}\mathbf{r1}_{3}^{*}\mathbf{r2}_{1}, \, \mathbf{0}\}. \\ (3.49)\\ \mathbf{r3}^{i} &= \mathbf{R}(\mathbf{r1}, \, \mathbf{r3}) = \{\mathbf{r3}_{0}, \, (\mathbf{1} - \mathbf{2}^{*}\mathbf{r1}_{0}^{2})^{*}\mathbf{r3}_{1}, \, - \, \mathbf{2}^{*}\mathbf{r1}_{0}^{*}\mathbf{r1}_{3}^{*}\mathbf{r3}_{1}, \, \mathbf{0}\}. \end{aligned}$$

Reflexión 1.- Reflexión de los eslabones 2 y 3 con Quaternión r21:

Para este caso observamos que las bases locales afectadas por la reflexión $r2^{i}$ son las que corresponden a los eslabones 2 y 3 (f_{i} y g_{i}). (Figura 3.17)

La reflexión anterior afecta el eje de giro del tercer eslabón, que al igual que en el caso anterior, tomaremos el sentido negativo del eje de rotación que es $-g_1^{1V}$, por ello el Quaternión $r3^1$ se transforman de la siguiente forma:

 $r3^{II} = \{r3_{0}, r3^{II}_{\nu}\};$ $r3^{II}_{\nu} = Cos(\theta_{3}/2)^{*} \cdot g_{1}^{IV} = Cos(\theta_{3}/2)^{*} R(r2^{I}, -g_{1}^{III}) = Cos(\theta_{3}/2)^{*} R(r2^{I}, (-\rho(r1, g_{1}^{II})))$ $r3^{II}_{\nu} = R(r2^{I}, -Cos(\theta_{3}/2)^{*} \rho(r1, g_{1}^{II})) = R(r2^{I}, r3^{I}_{\nu}) \implies r3^{II} = R(r2^{I}, r3^{I})$ $r3^{II} = R(r2^{I}, r3^{I}) = R(\rho(r1, r2), R(r1, r3)) = R(r1^{*}r\overline{2}^{*}r\overline{1}, r1^{*}r\overline{3}^{*}r\overline{1})$ $r3^{II} = r1^{*}r\overline{2}^{*}r\overline{1}^{*}r\overline{1}^{*}r\overline{1}^{*}r\overline{1}^{*}r\overline{1}^{*}r\overline{1}^{*}r\overline{2}^{*}r\overline{1} = r1^{*}r\overline{2}^{*}r\overline{1}^{*}r\overline{1}^{*}r\overline{1}^{*}r\overline{1}^{*}r\overline{2} = \rho((r1^{*}r\overline{2}), r3)$ (3.51)

Ahora, después de aplicadas las rotaciones anteriores el Quaternión que caracteriza el giro del tercer estabón es r3^{II}, y su forma explicita es:

$$\mathbf{r3}^{11} = \mathbf{R}(\mathbf{r2}^{1}, \mathbf{r3}^{1}) = \{\mathbf{r3}_{0}, (-1 + 2^{*}\mathbf{r1}_{0}^{2})^{*}\mathbf{r3}_{1}, 2^{*}\mathbf{r1}_{0}^{*}\mathbf{r1}_{3}^{*}\mathbf{r3}_{1}, 0\}.$$
 (3.52)

Reflexión 2.- Reflexión del tercer eslabón 3 con Quaternión r3":

Por último se tiene la reflexión del tercer eslabón con el Quaternión r3^{II}, con lo que se modifica la base local del eslabón 3. (Figura 3.18).

$$\mathbf{g}_{j}^{V} = \rho(\mathbf{r}\mathbf{3}^{H}, \mathbf{g}_{j}^{IV})$$
 (Reflexión del tercer eslabón con Quaternión r $\mathbf{3}^{H}$) (3.53)
Donde: $\mathbf{i} = 1.2.3$

Finalmente las bases que representan a cada eslabón son:

Eslabón 1:

$$e_{i} = \rho(r1, e_{i})$$
 (3.54)

Eslabón 2:

$$\begin{split} f_{j}^{(II)} &= \rho(r2^{I}, f_{j}^{(II)}) = R(R(r1, r2), \rho(r1, f_{j}^{(I)})) = R(r1 * r2 * r1, \rho(r1, \rho(P, f_{j}))) \\ f_{j}^{(II)} &= R(r1 * r2 * r1, r1 * P * r_{j} * P * r1) = -r1 * r2 * r1 * r1 * P * r_{j} * P * r1 * r1 * r2 * r1 \\ f_{j}^{(II)} &= -r1 * r2 * r1 * r1 * P * \overline{r_{j}} * P * r1 * r1 * r2 * r1 = -r1 * r2 * P * \overline{r_{j}} * P * r2 * r1 \\ f_{j}^{(II)} &= -r1 * r2 * P * \overline{r_{j}} * P * r2 * r1 = -r1 * r2 * P * \overline{r_{j}} * r1 * r2 * r1 \\ f_{j}^{(II)} &= -r1 * r2 * P * \overline{r_{j}} * P * r2 * r1 = -r1 * r2 * P * \overline{r_{j}} * r1 * r2 * P \\ \end{split}$$

$$f_{j}^{III} = R(r1^{*}r2^{*}P, f_{j})$$
 (3.55)

Eslabón 3:

$$g_{j}^{V} = R(r3^{n}, g_{j}^{|V|})$$

$$r3^{n} = r1^{*}r2^{*}r3^{*}r2^{*}r1$$

$$g_{j}^{|V|} = R(r2^{i}, g_{j}^{|I|}) = R(r1^{*}r2^{*}r1, \rho(r1, \rho(P,\rho(Q, g_{j}))))$$

$$g_{j}^{|V|} = R(r1^{*}r2^{*}r1, r1^{*}P^{*}Q^{*}g_{j}^{*}\overline{Q}^{*}\overline{P}^{*}r1) = r1^{*}r2^{*}r1^{*}r1^{*}P^{*}Q^{*}g_{j}^{*}\overline{Q}^{*}\overline{P}^{*}r\overline{1}^{*}r1^{*}r2^{*}r\overline{1}$$

$$g_{j}^{|V|} = r1^{*}r2^{*}r\overline{1}^{*}r1^{*}P^{*}Q^{*}\overline{g_{j}}^{*}\overline{Q}^{*}\overline{P}^{*}r\overline{1}^{*}r1^{*}r2^{*}r\overline{1} = r1^{*}r\overline{2}^{*}P^{*}Q^{*}\overline{g_{j}}^{*}\overline{Q}^{*}\overline{P}^{*}r2^{*}r\overline{1}$$

$$g_{j}^{|V|} = r1^{*}r\overline{2}^{*}P^{*}Q^{*}\overline{g_{j}}^{*}\overline{Q}^{*}\overline{P}^{*}r2^{*}r\overline{1}$$

Sustituyendo r3" y $\;g_{j}^{1V}\;$ en $\;g_{j}^{V}\;$, tenemos que:

$$\mathbf{g}_{j}^{V} = r\mathbf{1}^{*}\mathbf{r}\mathbf{\overline{2}}^{*}\mathbf{r}\mathbf{3}^{*}\mathbf{P}^{*}\mathbf{Q}^{*}\mathbf{g}_{j}^{*}\mathbf{\overline{Q}}^{*}\mathbf{\overline{P}}^{*}\mathbf{r}\mathbf{\overline{3}}^{*}\mathbf{r}\mathbf{2}^{*}\mathbf{r}\mathbf{\overline{1}} = r\mathbf{1}^{*}\mathbf{r}\mathbf{\overline{2}}^{*}\mathbf{r}\mathbf{3}^{*}\mathbf{P}^{*}\mathbf{Q}^{*}\mathbf{g}_{j}^{*}\mathbf{r}\mathbf{\overline{1}}^{*}\mathbf{r}\mathbf{\overline{2}}^{*}\mathbf{r}\mathbf{3}^{*}\mathbf{P}^{*}\mathbf{Q}$$

$$g_{j}^{V} = \rho(r1^{+}r2^{+}r3^{+}P^{+}Q, g_{j})$$
 (3.56)

3.2.2. Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada.

De las figuras Figura 3.18 y Figura 3.19 se determinan los vectores que definen los eslabones del manipulador y el vector del posición del elemento terminal (punto p). Las longitudes de los eslabones son L_1 , L_2 , L_3 .

$$b_1 = L_1^* e_3^1$$
; (Eslabón 1)
 $b_2 = L_2^* f_3^{111}$; (Eslabón 2) (3.57)
 $b_3 = L_3^* g_3^V$; (Eslabón 3)

$$r_p = b_1 + b_2 + b_3$$
 (Posición del elemento terminal) (3.58)

3.2.3. Ecuaciones explicitas optimizadas para la configuración deformada.

b ₁ = {0, 0, L ₁ };	
b ₂ = {4*A5*B2*L ₂ , 2*A6*B2*L ₂ , - B1*L ₂ };	(3.59)
$b_3 = \{4^*A6^*C1^*L_3, 2^*A6^*C1^*L_3, C2^*L_3\};$	

$$\mathbf{r}_{p} = \{\mathbf{4}^{*}\mathbf{A5}^{*}(\mathbf{B2}^{*}\mathbf{L}_{2} + \mathbf{C1}^{*}\mathbf{L}_{3}), \mathbf{2}^{*}\mathbf{A6}^{*}(\mathbf{B2}^{*}\mathbf{L}_{2} + \mathbf{C1}^{*}\mathbf{L}_{3}), \mathbf{L}_{1} - \mathbf{B1}^{*}\mathbf{L}_{2} + \mathbf{C2}^{*}\mathbf{L}_{3}\}; \quad (3.60)$$

.

Donde:

MODELACIÓN CINEMATICA



Figura 3.19. Vector de posición del elemento terminal en la configuración deformada.

Utilizando los dos modelos cinemáticos anteriores de manera conjunta, se puede hacer que el manipulador siga una trayectoria determinada evitando algunos obstáculos, al utilizar un modelo u otro, dependiendo de la ubicación del obstáculo en el espacio. Esto se verá en las simulaciones cinemáticas que se mostraran más adelante.

3.3. MODELACIÓN CINEMÁTICA DE UN MANIPULADOR DE CUATRO GRADOS DE LIBERTAD, UTILIZANDO ROTACIONES.

En esta sección mostraremos la modelación de un manipulador de cuatro grados de libertad, aplicando la secuencia más sencilla determinada en la modelación del manipulador de tres grados de libertad, tanto para la configuración deformada como para la no deformada.

3.3.1. Configuración no deformada.

Mostraremos primeramente la modelación de la configuración no deformada, que consiste en determinar un sistema de ecuaciones que nos permitan determinar la posición del segundo y tercer eslabón, cuando el primer eslabón se encuentra en una posición inicial de referencia (home), conociendo la posición del extremo del tercer eslabón, que consideraremos para este caso como la posición del elemento terminal del manipulador.

Secuencia de Rotaciones:

Se mantiene fijo el el cuarto eslabón.
 Rotación 2.- Se aplica una rotación al tercer eslabón. (Figura 3.21).
 Rotación 3.- Se aplica una rotación al segundo eslabón. (Figura 3.22).
 4.- Se mantiene fijo el primer eslabón.

Para lo anterior definiremos la configuración no deformada del manipulador, que consiste en definir el vector de posición del elemento terminal para posiciones arbitrarias del segundo y tercer eslabones en un plano dado. Consideremos la base canónica de \Re^3 , esto es:

$$e_1 = \{1,0,0\}; e_2 = \{0,1,0\}; e_3 = \{0,0,1\};$$

Ubicamos sistemas de referencia locales en la base de cada eslabón del manipulador como se muestra en la Figura 3.20. Observemos que, en este caso, los sistemas de referencia definidos coinciden con la base canónica la cual sirve como sistema de referencia inercial fijo, es decir:

$$h_i = g_i = f_i = e_i$$
, $j = 1, 2, 3$.



Figura 3.20. Bases locales en la configuración original para la modelación.

De la posición mostrada en la Figura 3.20, debemos permitir que los eslabones 2 y 3 giren a una posición arbitraria, considerando que los eslabones tienen los siguientes ejes de rotación:

Eslabón	Eje De Rotación
1	e,
2	$\mathbf{f}_1 = \mathbf{e}_1$
3	$\mathbf{g}_1 = \mathbf{e}_1$
4	$h_3 = e_3$

Con los ejes de rotación anteriores, podemos definir la forma que tendrán inicialmente los Quaterniones que nos permitirán rotar los eslabones 2 y 3.

Estabón 2: Consideremos el siguiente parámetro que caracterizará a nuestra primera rotación;

 $P = \{P_0, P_1, P_2, P_3\} \implies P_v = \{P_1, P_2, P_3\}, P = 1.$

En este caso se tiene que si $\underline{w} \in \Re^3$ es el eje de rotación, entonces, [1]:

$$\begin{split} & \mathsf{P}_{0} = \mathsf{Cos}(\varphi_{1}/2) \\ & \mathsf{P}_{v} = \mathsf{Sin}(\varphi_{1}/2)^{*} \underline{w} = \mathsf{Sin}(\varphi_{1}/2)^{*} \mathbf{f}_{1} = \mathsf{Sin}(\varphi_{1}/2)^{*} \mathbf{e}_{1} = \mathsf{Sin}(\varphi_{1}/2)^{*} \{1,0,0\} \\ & \mathsf{P}_{v} = \{\mathsf{P}_{1},0,0\} \end{split}$$

De lo anterior se obtiene la forma local del parámetro P, que es: P = {Po, P1, 0, 0}, Donde:

 $P_0 = Cos(\phi_1/2);$ $P_1 = Sen(\phi_1/2);$

Siendo ϕ_1 el ángulo de rotación del eslabón 2.

Estabón 3: Consideremos ahora el siguiente parámetro que caracterizará la segunda rotación;

 $Q = \{Q_0, Q_1, Q_2, Q_3\} \implies Q_v = \{Q_1, Q_2, Q_3\}, ||Q|| = 1.$

Siendo $w \in \Re^3$ el eje de rotación, tenemos que:

 $\begin{aligned} Q_v &= \operatorname{Sin}(\phi_2/2)^* \underline{w} = \operatorname{Sin}(\phi_2/2)^* g_1 = \operatorname{Sin}(\phi_2/2)^* e_1 = \operatorname{Sin}(\phi_2/2)^* \{1,0,0\} \\ Q_v &= \{Q_1, Q, Q\} \end{aligned}$

De lo anterior se obtiene la forma local del parámetro Q, que es: Q = (Q₀, Q₁, 0, 0), Donde:

 $Q_0 = Cos(\phi_2/2);$ $Q_1 = Sen(\phi_2/2);$

Siendo φ_2 el ángulo de rotación del eslabón 3.

Con los Quaterniones anteriores podemos ahora realizar las rotaciones necesarias para determinar la configuración no deformada, se debe observar que si los ejes de rotación cambian su forma, también cambiará la forma de los Quaterniones dados.

Aplicación de la Secuencia de rotaciones para determinar la configuración no deformada.

Rotación 2.- Rotamos el tercer eslabón con el Quaternión Q. Esto se realiza rotando las bases locales g_i y h_i . (Figura 3.21)

- $\mathbf{g}_{i}^{T} = \rho(\mathbf{Q}, \mathbf{g}_{i})$ (Rotación del tercer eslabón con Quaternión Q), j = 1, 2, 3
- $\mathbf{h}_{i}^{1} = \rho(\mathbf{Q}, \mathbf{h}_{i})$ (Rotación del cuarto eslabón con Quaternión Q), j = 1,2.3 (3.62)



Figura 3.21. Rotación del tercer eslabón en la configuración no deformada.

Rotación 3.- Se aplica la rotación al segundo eslabón con el Quaternión P, lo que implica que las bases f_i , g_i y h_i . sufren las siguientes transformaciones.(Figura 3.22)

$$\begin{array}{l} f_{j}^{I} = \rho(P, f_{j}) & (\text{Rotación del segundo eslabón con Quaternión P}) \\ g_{j}^{II} = \rho(P, g_{j}^{I}) & (\text{Rotación del tercer eslabón con Quaternión P}). \\ h_{j}^{II} = \rho(Q, h_{j}^{I}) & (\text{Rotación del cuarto eslabón con Quaternión P}), \ j = 1,2,3 \end{array}$$

Como $\mathbf{g}_i^{\mathbf{I}} = \rho(\mathbf{Q}, \mathbf{g}_i) \text{ y } \mathbf{h}_i^{\mathbf{I}} = \rho(\mathbf{Q}, \mathbf{h}_i)$, tenemos que:

$$g_{j}^{11} = \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{g}_{j})) = \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{g}_{j})$$
(3.64)
$$\mathbf{h}_{j}^{11} = \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{h}_{j})) = \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{h}_{j})$$

MODELACION CINEMATICA



Figura 3.22. Rotación de los eslabones 2 y 3 en la configuración no deformada.

La forma explícita de las bases locales, después de aplicadas las rotaciones anteriores, es la siguiente:

$$\begin{aligned} f_1^{l} &= \{1, 0, 0\} \\ f_2^{l} &= \{0, -1 + 2^* P_0^{2}, 2^* P_0^{*} P_1\} \\ &= \{0, -2^* P_0^{*} P_{11}, -1 + 2^* P_0^{2}\} \\ g_1^{ll} &= \{1, 0, 0\} \\ g_2^{ll} &= \{0, -P_0^{2} + P_1^{2} + 2^* P_0^{2*} Q_0^{2} - 2^* P_1^{2*} Q_0^{2} - 4^* P_0^{*} P_1^{*} Q_0^{*} Q_1, \\ &= 2^* (-\{P_0^{*} P_1\}) + 2^* P_0^{*} P_1^{*} Q_0^{2} + P_0^{2*} Q_0^{*} Q_1 - P_1^{2*} Q_0^{*} Q_1)\} \\ g_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_1^{*} Q_0^{2} - P_0^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1)\} \\ h_1^{ll} &= \{1, 0, 0\} \\ h_2^{ll} &= \{0, -P_0^{2} + P_1^{2} + 2^* P_0^{2*} Q_0^{2} - 2^* P_1^{2*} Q_0^{2} - 4^* P_0^{*} P_1^{*} Q_0^{*} Q_1\} \\ h_1^{ll} &= \{1, 0, 0\} \\ h_2^{ll} &= \{0, -P_0^{2} + P_1^{2} + 2^* P_0^{2*} Q_0^{2} - 2^* P_1^{2*} Q_0^{2} - 4^* P_0^{*} P_1^{*} Q_0^{*} Q_1\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_1^{*} Q_0^{2} - P_0^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1)\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_1^{*} Q_0^{2} - P_0^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1)\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_1^{*} Q_0^{2} - P_0^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1)\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_1^{*} Q_0^{2} - 2^* P_1^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1)\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_1^{*} Q_0^{2} - 2^* P_1^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1)\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_1^{*} Q_0^{2} - 2^* P_1^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1)\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_1^{*} Q_0^{2} - 2^* P_1^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1)\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_1^{*} Q_0^{2} - 2^* P_1^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1)\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_1^{*} Q_0^{2} - 2^* P_1^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1)\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_0^{*} Q_0^{2} - 2^* P_1^{2*} Q_0^{*} Q_1 + P_1^{2*} Q_0^{*} Q_1\} \\ h_3^{ll} &= \{0, 2^* (P_0^{*} P_1 - 2^* P_0^{*} P_0^{*} Q_0^{2} - 2^* P_1^{2*} Q_0^{*} Q_0^{*} + P_0^{2*} Q_0^{*} Q_1\}$$

3.3.2. Vectores que definen los eslabones y posición del elemento terminal en la configuración no deformada.

De la Figura 3.22 y Figura 3.24 podemos determinar los vectores que definen los eslabones del manipulador y el vector de posición del elemento terminal (punto p). Siendo L_1 , L_2 , L_3 , L_4 y L_5 las longitudes de los eslabones.

$$\begin{aligned} \mathbf{b}_1 &= \mathbf{L}_1^* \mathbf{e}_3; \quad (\text{Eslabon 1}) \\ \mathbf{b}_2 &= \mathbf{L}_2^* \mathbf{f}_3^1 = \mathbf{L}_2^* \rho(\mathbf{P}, \mathbf{f}_3) = \mathbf{L}_2^* (\mathbf{P}^* \mathbf{e}_3^* \mathbf{\overline{P}}); \quad (\text{Eslabon 2}) \\ \mathbf{b}_3 &= \mathbf{L}_3^* \mathbf{g}_3^{11} = \mathbf{L}_3^* \rho((\mathbf{P}^* \mathbf{Q}), \mathbf{g}_3) = \mathbf{L}_3^* ((\mathbf{P}^* \mathbf{Q})^* \mathbf{e}_3^* (\mathbf{\overline{P}^* \mathbf{Q}})); \quad (\text{Eslabon 3}) \end{aligned}$$
(3.68)

$$\mathbf{R}_{p} = \mathbf{b}_{1} + \mathbf{b}_{2} + \mathbf{b}_{3} \quad (\text{Posición del elemento terminal}) \quad (3.69)$$

De la Figura 3.22 podemos observar que la base local \mathbf{h}_{j}^{H} es la que define la pinza o eslabón 4 del manipulador, de la misma figura se puede ver que son necesarios cuando menos dos vectores para definir dicha pinza, uno paralelo y otro perpendicular a \mathbf{b}_{3} , lo cual se muestra en la siguiente figura.



Figura 3.23. Vectores que definen la pinza del manipulador (cuarto eslabón).

De lo anterior tenemos que los vectores que definen la pinza son los siguientes.

$$\begin{aligned} \mathbf{b}_4 &= \mathbf{L}_4^* \mathbf{h}_3^{11} = \mathbf{L}_4^* \rho((\mathbf{P}^* \mathbf{Q}), \mathbf{h}_3) = ((\mathbf{P}^* \mathbf{Q})^* \mathbf{e}_3^* (\overline{\mathbf{P}^* \mathbf{Q}})); \\ \mathbf{b}_8 &= \mathbf{L}_8^* \mathbf{h}_1^{11} = \mathbf{L}_8^* \rho((\mathbf{P}^* \mathbf{Q}), \mathbf{h}_1) = ((\mathbf{P}^* \mathbf{Q})^* \mathbf{e}_1^* (\overline{\mathbf{P}^* \mathbf{Q}})); \end{aligned}$$
(3.70)

MODELACION CINEMATICA



Figura 3.24. Vector de posición del elemento terminal en la configuración no deformada.

3.3.3. Ecuaciones explícitas optimizadas para la configuración no deformada.

A continuación se muestra la forma de las ecuaciones optimizadas, correspondientes a la configuración no deformada.

Donde:

A1 =
$$P_0^*P_1$$
; A2 = 1 - 2* P_0^2 ;
A3 = $Q_0^*Q_1$; A4 = 1 - 2* Q_0^2 ; (3.74)

3.3.4. Configuración deformada.

Ahora se modelará la configuración deformada, iniciando las rotaciones en el elemento terminal y finalizando en el primer elemento, ya que es la secuencia de modelación más sencilla.

Secuencia de Rotaciones:

Rotación 1.- Rota el eslabón 4 con el Quaternión r4, partiendo de la configuración no deformada. (Figura 3.25).



Figura 3.25. Rotación del cuarto eslabón en la configuración deformada.

Rotación 2.- Rotan los eslabones 3 y 4 con el Quaternión r3. (Figura 3.26).



Figura 3.26. Rotación de los eslabones 3 y 4 en la configuración deformada.

Rotación 3.- Rotan los eslabones 2, 3 y 4 con el Quaternión r2. (Figura 3.28).



Figura 3.27. Rotación de los eslabones 2, 3 y 4 en la configuración deformada.

Rotación 3.- Rotan los cuatro eslabones con el Quaternión r1. (Figura 3.28).



Figura 3.28. Rotación de los cuatro eslabones en la configuración deformada.

De la Figura 3.25 podemos ver que los ejes de rotación de los eslabones se conservan como en la configuración original (Figura 3.20), excepto para el cuarto eslabón, cuyo eje de rotación es $h_3^{\rm II}$. Por lo anterior podemos definir los Quaterniones para rotar a posición deformada como se muestra a continuación.

Eslabón 1: $r1 = \{r1_0, r1_1, r1_2, r1_3\} \implies r1_v = \{r1_1, r1_2, r1_3\}, ||r1|| = 1.$

La forma local del Quaternión r1 es:
De lo anterior: r1 = { r10, 0, 0, r13 }, Donde:

 $r1_0 = Cos(\theta_1/2);$ $r1_3 = Sen(\theta_1/2);$

Siendo 0, el ángulo de rotación del eslabón 1 en la configuración deformada.

Eslabón 2: $r2 = \{r2_0, r2_1, r2_2, r2_3\} \implies r2_v = \{r2_1, r2_2, r2_3\}, ||r2|| = 1.$

La forma local del Quaternión r2 es:

 $r_2 = Sin(\theta_2/2)^* e_1 = Sin(\theta_2/2)^* \{1, 0, 0\}$ $r_2 = \{r_2, 0, 0\}$

De lo anterior: r2 = { r20, r21, 0, 0}, Donde:

 $r_{2_0} = Cos(\theta_2/2);$ $r_{2_1} = Sen(\theta_2/2);$

Siendo 02 el ángulo de rotación del eslabón 2 en la configuración deformada.

Eslabón 3: $r3 = \{r3_0, r3_1, r3_2, r3_3\} \implies r3_v = \{r3_1, r3_2, r3_3\}, ||r3|| = 1.$

La forma local del Quaternión r3 es:

 $r_{3_v} = Sin(\theta_3/2)^* e_1 = Sin(\theta_3/2)^* \{1, 0, 0\}$ $r_{3_v} = \{r_{3_1}, 0, 0\}$

De lo anterior: r3 = { r3,, r3, 0, 0 }, Donde:

 $r_{3_0} = Cos(\theta_3/2);$ $r_{3_1} = Sen(\theta_3/2);$

Siendo θ₃ el ángulo de rotación del eslabón 3 en la configuración deformada.

Eslabón 4: $r4 = \{r4_0, r4_1, r4_2, r4_3\} \implies r4_v = \{r4_1, r4_2, r4_3\}, ||r4|| = 1.$

La forma local del Quaternión r4 en la configuración no deformada es:

$$r4_v = Sin(\theta_4/2)^* e_3 = Sin(\theta_4/2)^* \{0, 0, 1\}$$

 $r4_v = \{0, 0, r4_3\}$

De lo anterior: r4 = { r40, 0, 0, r43 }, Donde:

 $r4_0 = Cos(\theta_4/2);$ $r4_3 = Sen(\theta_4/2);$

Siendo θ_4 el ángulo de rotación del eslabón 4 en la configuración deformada.

Pero como el eje de rotación del eslabón 4 se ve afectado por las rotaciones aplicadas en la configuración no deformada, siendo ahora $h_3^{\rm II}$, el Quaternión r4, será:

$$r\mathbf{4}_{v}^{I} = \operatorname{Sin}(\theta_{4}/2)^{*} \mathbf{h}_{3}^{II} = \operatorname{Sin}(\theta_{4}/2)^{*}\rho((\mathsf{P}^{*}\mathsf{Q}), \mathbf{e}_{3}) = \rho((\mathsf{P}^{*}\mathsf{Q}), r\mathbf{4}_{v})$$
$$r\mathbf{4}^{I} = \rho((\mathsf{P}^{*}\mathsf{Q}), r\mathbf{4})$$

Aplicación de la Secuencia de rotaciones para determinar la configuración deformada.

Rotación 1.- Rotación del cuarto eslabón con el Quaternión r4. Esta rotación solamente afecta al eslabón 4. (Figura 3.25).

$$\mathbf{h}_{j}^{III} = \rho(\mathbf{r4}^{I}, \mathbf{h}_{j}^{II})$$
 (Rotación del cuarto eslabón con Quaternión r4)
Donde: $\mathbf{j} = 1,2,3$ (3.75)

Rotación 2.- Rotación de los eslabones 3 y 4 con el Quaternión r3. Esta rotación afecta las bases g, y h, (Figura 3.26).

$\mathbf{g}_{j}^{iii} = \rho(\mathbf{r}3, \mathbf{g}_{j}^{ii})$	(Rotación del tercer eslabón con Quaternión r3)	
$\mathbf{h}_{j}^{\mathrm{IV}} = \rho(\mathbf{r3}, \mathbf{h}_{j}^{\mathrm{III}})$	(Rotación del cuarto eslabón con Quaternión r3)	(3.76)
Donde: j = 1,2,3		

Rotación 3.- Rotación de los eslabones 2, 3 y 4 con el Quaternión r2. Esta rotación afecta las bases f_j , g_j y h_j . (Figura 3.27).

$f_{j}^{11} = \rho(r2, f_{j}^{1})$	(Rotación del segundo eslabón con Quaternión r2)	
$\mathbf{g}_{j}^{\mathrm{IV}} = \rho(\mathbf{r2}, \mathbf{g}_{j}^{\mathrm{III}})$	(Rotación del tercer eslabón con Quaternión r2)	(3.77)
$\mathbf{h}_{j}^{\mathbf{V}} = \rho(\mathbf{r2}, \mathbf{h}_{j}^{\mathbf{IV}})$	(Rotación del cuarto eslabón con Quaternión r2)	
Donde: j = 1,2,3	,	

Rotación 4.- Rotación de los cuatro eslabones con el Quaternión r1. Esta rotación, que es la última, afecta a todas las bases.. (Figura 3.28).

$\mathbf{e}_{j}^{\dagger} = \rho(\mathbf{r1}, \mathbf{e}_{j})$	(Rotación del primer eslabón con Quaternión r1)	
$\mathbf{f}_{j}^{III} = \rho(\mathbf{r1}, \mathbf{f}_{j}^{II})$	(Rotación del segundo eslabón con Quaternión r1)	
$\mathbf{g}_{\mathbf{j}}^{\mathbf{V}} = \rho(\mathbf{r1}, \mathbf{g}_{\mathbf{j}}^{\mathbf{W}})$	(Rotación del tercer eslabón con Quaternión r1)	(3.78)
$h_{j}^{V1} = \rho(r1, h_{j}^{V})$	(Rotación del cuarto eslabón con Quaternión r1)	
Donde: j = 1,2,3		

La forma final resultante de las bases locales de cada eslabón es la siguiente:

Eslabón 1.-

$$e_{j} = \rho(r1, e_{j})$$
 (3.79)

Eslabón 2.-

$$\begin{aligned} \mathbf{f}_{j}^{III} &= \rho(\mathbf{r1}, \mathbf{f}_{j}^{II}) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \mathbf{f}_{j}^{I})) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{P}, \mathbf{f}_{j}))) = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{f}_{j}) \\ \mathbf{f}_{j}^{III} &= \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{P}, \mathbf{e}_{j}))) = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{e}_{j}) \end{aligned}$$
(3.80)

Eslabón 3.-

$$\begin{aligned} \mathbf{g}_{j}^{V} &= \rho(\mathbf{r1}, \mathbf{g}_{j}^{|V|}) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \mathbf{g}_{j}^{||1|})) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{r3}, \mathbf{g}_{j}^{||1|}))) \\ \mathbf{g}_{j}^{V} &= \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{r3}, \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{e}_{j}))))) = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{e}_{j}) \end{aligned}$$
(3.81)

Eslabón 4.-

 $\mathbf{h}_{j}^{VI} = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{r3}, \rho(\mathbf{P}, \rho(\mathbf{Q}, \rho(\mathbf{r4}, \mathbf{h}_{j}))))) = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}^{*}\mathbf{r4}, \mathbf{e}_{j})$ (3.82)

3.3.5. Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada.

De la Figura 3.29 y Figura 3.29 se determinan los vectores que definen los eslabones del manipulador y el vector del posición del elemento terminal (punto p), que en este caso, se considera ser un punto central del extremo de la pinza. Siendo L₁, L₂, L₄ y L₅ las longitudes de los eslabones.

$$\mathbf{r}_p = \mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3 + \mathbf{b}_4$$
 (Posición del elemento terminal) (3.84)



Figura 3.29. Vector de posición del elemento terminal en la configuración no deformada.

3.3.6. Ecuaciones explícitas optimizadas para la configuración deformada.

Las expresiones anteriores se explicitaron y optimizaron utilizando Mathematica®, y se presentan a continuación.

$$b_1 = \{0, 0, L_1\}$$

$$b_2 = \{-4^A5^B2^L_2, -2^A6^B2^L_2, B1^L_2\}$$

$$b_3 = \{-4^A5^C1^L_3, -2^A6^C1^L_3, C2^L_3\}$$

$$b_4 = \{-4^A5^C1^L_4, -2^A6^C1^L_4, C2^L_4\}$$

$$b_5 = \{(A12^A6 - 4^A11^A5^C2)^L_5, -2^A(A12^A5 + A11^A6^C2)^L_5, -4^A11^C1^L_5\};$$

$$r_{p} = \{-4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}(L_{3}+L_{4})), \\ -2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}(L_{3}+L_{4})), L_{1} + B1^{*}L_{2} + C2^{*}(L_{3}+L_{4})\}$$
(3.86)

Donde:

A1 =
$$P_0^*P_1$$
; A2 = $1 \cdot 2^*P_0^2$;
A3 = $Q_0^*Q_1$; A4 = $1 \cdot 2^*Q_0^2$;
A5 = $r1_0^*r1_3$; A6 = $1 \cdot 2^*r1_0^2$;
A7 = $r2_0^*r2_1$; A8 = $1 \cdot 2^*r2_0^2$; (3.87)
A9 = $r3_0^*r3_1$; A10 = $1 \cdot 2^*r3_0^2$;
A11 = $r4_0^*r4_3$; A12 = $1 \cdot 2^*r4_0^2$;
B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8;
B3 = A10*A8 - 4*A1*A7; B4 = A10*A7 + A8*A9;
B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4;
C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6;

3.4. MODELACION CINEMATICA DE UN MANIPULADOR DE CUATRO GRADOS DE LIBERTAD TIPO SCARA, UTILIZANDO ROTACIONES.

En esta sección mostraremos la modelación de un robot tipo SCARA de cuatro grados de libertad, aplicando la secuencia más sencilla determinada en la modelación del manipulador de tres grados de libertad, tanto para la configuración deformada como para la no deformada...

3.4.1. Configuración no deformada.

Mostraremos primeramente la modelación de la configuración no deformada, que consiste en determinar un sistema de ecuaciones que nos permitan determinar la posición del primero, segundo y tercer eslabón, conociendo la posición del elemento terminal del manipulador. Para determinar las ecuaciones necesarias realizamos las siguientes rotaciones y una traslación.

Se mantiene fija la pinza.
 Traslación 2.- Se aplica una traslación al tercer eslabón. (Figura 3.31).
 Rotación 3.- Se aplica una rotación al segundo eslabón. (Figura 3.32).
 Rotación 4.- Se aplica una rotación al primer eslabón. (Figura 3.33).

Para lo anterior definiremos la configuración no deformada del manipulador, que consiste en definir el vector de posición del elemento terminal para posiciones arbitrarias del primero y segundo eslabón. Consideremos la base canónica de \Re^3 , esto es:

 $e_1 = \{1,0,0\}; e_2 = \{0,1,0\}; e_3 = \{0,0,1\};$

Ubicamos sistemas de referencia locales en la base de cada eslabón del manipulador como se muestra en la Figura 3.30. Observemos que, en este caso, los sistemas de referencia definidos coinciden con la base canónica la cual sirve como sistema de referencia inercial fijo, es decir:

$$h_i = g_i = f_i = e_i$$
, $j = 1,2,3$.



Figura 3.30. Bases locales en la configuración original para la modelación.

De la posición mostrada en la Figura 3.30, debemos permitir que los eslabones 1 y 2 giren a una posición arbitraria, considerando que los eslabones tienen los siguientes ejes de rotación o traslación:

Eslabón	Eje De Rotación
1	e,
2	$f_3 = e_3$
3	$g_3 = e_3$ (Traslación)
4	$h_3 = e_3$

Con los ejes de rotación anteriores, podemos definir la forma que tendrán inicialmente los Quaterniones que nos permitirán rotar los eslabones 1 y 2, y la variable a utilizar para la traslación del tercer eslabón..

Eslabón 1: Consideremos el siguiente parámetro que caracterizará a nuestra primera rotación;

 $P = \{P_0, P_1, P_2, P_3\} \Rightarrow P_v = \{P_1, P_2, P_3\}, P = 1.$

En este caso se tiene que siendo $w \in \Re^3$ el eje de rotación, entonces, [1]:

 $P_{0} = Cos(\phi_{1}/2)$ $P_{v} = Sin(\phi_{1}/2)^{*} \underline{w} = Sin(\phi_{1}/2)^{*}e_{3} = Sin(\phi_{1}/2)^{*}\{0,0,1\}$ $P_{v} = \{0, 0, P_{3}\}$

De lo anterior se obtiene la forma local del parámetro P, que es: P = {Po, 0, 0, P3}, Donde:

 $P_0 = Cos(\phi_1/2);$ $P_3 = Sen(\phi_1/2);$

Siendo ϕ_1 el ángulo de rotación del eslabón 1.

Eslabón 2: Consideremos ahora el siguiente parámetro que caracterizará la segunda rotación;

 $Q = \{Q_0, Q_1, Q_2, Q_3\} \Rightarrow Q_v = \{Q_1, Q_2, Q_3\}, ||Q|| = 1.$

Siendo $\underline{w} \in \Re^3$ el eje de rotación, tenemos que:

 $\begin{array}{l} Q_v = Sin(\phi_2/2)^* \underline{w} = Sin(\phi_2/2)^* f_3 = Sin(\phi_2/2)^* e_3 = Sin(\phi_2/2)^* \{0,0,1\} \\ Q_v = \{ \ 0, \ 0, \ Q_3 \} \end{array}$

De lo anterior se obtiene la forma local del parámetro Q, que es: Q = {Q₀, 0, 0, Q₃}, Donde:

 $Q_0 = Cos(\phi_2/2);$ $Q_3 = Sen(\phi_2/2);$

Siendo ϕ_2 el ángulo de rotación del eslabón 2.

Eslabón 3: Como este eslabón se traslada exclusivamente sobre el eje e_3 , su desplazamiento, en la configuración no deformada, lo definiremos simplemente como la magnitud de la variable $I_{3,..}$

Con los Quaterniones anteriores podemos ahora realizar las rotaciones necesarias para determinar la configuración no deformada, se debe observar que si los ejes de rotación cambian su forma, también cambiará la forma de los Quaterniones dados.

Aplicación de la Secuencia de rotaciones y traslaciones para determinar la configuración no deformada.

Traslación 2.- La aplicación de la traslación no afecta la forma de las bases locales de los eslabones, como se puede observar en Figura 3.31.



Figura 3.31. Traslación del eslabón 3 (I_{3v}) en la configuración no deformada.

Rotación 3.- Rotamos el segundo eslabón con el Quaternión Q. Esto se realiza rotando las bases locales f_i, g_i y h_i. (Figura 3.32)

$\mathbf{f}_j^1 = \rho(\mathbf{Q}, \mathbf{f}_j)$	(Rotación del segundo eslabón con Quaternión Q),	
$\mathbf{g}_{j}^{I} = \rho(\mathbf{Q}, \mathbf{g}_{j})$	(Rotación del tercer eslabón con Quaternión Q),	(3.88)
$h_j^1 = \rho(Q, h_j)$ j = 1,2,3	(Rotación del cuarto eslabón con Quaternión Q),	

MODELACION CINEMATICA



Figura 3.32. Rotación del segundo eslabón en la configuración no deformada.

Rotación 4.- Se aplica la rotación al primer eslabón con el Quaternión P, lo que implica que las bases e_j , f_j , g_j y h_j . sufren las siguientes transformaciones.(Figura 3.33)

$\mathbf{e}_{j}^{i} = \rho(\mathbf{P}, \mathbf{e}_{j})$	(Rotación del primer eslabón con Quaternión P).	
$\mathbf{f}_{j}^{\mathrm{II}} = \rho(\mathbf{P}, \mathbf{f}_{j}^{\mathrm{I}})$	(Rotación del segundo eslabón con Quaternión P).	
$\mathbf{g}_{j}^{II} = \rho(\mathbf{P}, \mathbf{g}_{j}^{I})$	(Rotación del tercer eslabón con Quaternión P).	(3.89)
$\mathbf{h}_{j}^{II} = \rho(\mathbf{Q}, \mathbf{h}_{j}^{I})$	(Rotación del cuarto eslabón con Quaternión P).	
j = 1,2,3		

Como $\mathbf{f}_{j}^{1} = \rho(\mathbf{Q}, \mathbf{f}_{j}), \ \mathbf{g}_{j}^{1} = \rho(\mathbf{Q}, \mathbf{g}_{j}) \ \mathbf{y} \ \mathbf{h}_{j}^{1} = \rho(\mathbf{Q}, \mathbf{h}_{j}), \text{ tenemos que:}$

$$\begin{aligned} \mathbf{f}_{j}^{II} &= \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{f}_{j})) = \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{f}_{j}) \\ \mathbf{g}_{j}^{II} &= \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{g}_{j})) = \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{g}_{j}) \end{aligned} (3.90) \\ \mathbf{h}_{i}^{II} &= \rho(\mathbf{P}, \rho(\mathbf{Q}, \mathbf{h}_{i})) = \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{h}_{i}) \end{aligned}$$

MODELACION CINEMATICA



Figure 3.33. Rotación de los eslabones 2 y 3 en la configuración no deformada.

La forma explicita de las bases locales, después de aplicadas las rotaciones anteriores, es la siguiente:

$$e_{1}^{i} = \{-1 + 2^{*}P_{0}^{2}, 2^{*}P_{0}^{*}P_{3}, 0\}$$

$$e_{2}^{i} = \{-2^{*}P_{0}^{*}P_{3}, -1 + 2^{*}P_{0}^{2}, 0\}$$

$$(3.91)$$

$$e_{3}^{i} = \{0, 0, 1\}$$

$$f_{1}^{ii} = \{1 - 2^{*}P_{0}^{2} - 2^{*}Q_{0}^{2} + 4^{*}P_{0}^{2*}Q_{0}^{2} - 4^{*}P_{0}^{*}P_{3}^{*}Q_{0}^{*}Q_{3}, -2^{*}P_{0}^{*}P_{3} + 4^{*}P_{0}^{*}P_{0}^{*}Q_{0}^{2} - 2^{*}Q_{0}^{*}Q_{3} + 4^{*}P_{0}^{2*}Q_{0}^{*}Q_{3}, 0\}$$

$$f_{2}^{ii} = \{2^{*}(P_{0}^{*}P_{3} - 2^{*}P_{0}^{*}P_{3}^{*}Q_{0}^{2} + Q_{0}^{*}Q_{3} - 2^{*}P_{0}^{2*}Q_{0}^{*}Q_{3}, 0\}$$

$$f_{3}^{ii} = \{0, 0, 1\}$$

$$h_{1}^{ii} = g_{1}^{ii} = f_{1}^{ii}$$

$$h_{2}^{ii} = g_{1}^{ii} = f_{2}^{ii}$$

$$(3.93)$$

75

3.4.2. Vectores que definen los eslabones y posición del elemento terminal en la configuración no deformada.

De la Figura 3.33 y Figura 3.34 podemos determinar los vectores que definen los eslabones del manipulador y el vector de posición del elemento terminal (punto p). Siendo L_1 , L_2 y L_3 las longitudes de los eslabones.

 $\begin{array}{ll} b_1 = L_1^* e_1^1 = L_2^* \; \rho(P, e_1) & ; & (\; \text{Eslabón 1} \;) \\ b_2 = L_2^* f_1^{11} = L_2^* \; \rho(P, \; f_1^1) = L_2^* \; \rho(P_1 \rho(Q, \; f_1)); & (\; \text{Eslabón 2} \;) \\ b_3 = L_3^* (-g_3^{11}) = -L_3^* \; e_3; & (\; \text{Eslabón 3 fijo}) & (\; 3.94 \;) \\ b_{3v} = l_{3v}^* (-g_3^{11}) = -l_{3v}^* \; e_3; & (\; \text{Eslabón 3 variable}) \\ b_4 = L_4^* (-h_3^{11} \;); & (\; \text{Pinza} \;) \\ b_5 = L_5^* \; h_1^{11} \; ; & (\; \text{Pinza}) \\ \end{array}$

R_p = b₁ + b₂ + b₃ + b₃, (Posición del elemento terminal) (3.95)

Figura 3.34. Vector de posición del elemento terminal en la configuración no deformada.

3.4.3. Ecuaciones explícitas optimizadas para la configuración no deformada.

A continuación se muestra la forma de las ecuaciones optimizadas, correspondientes a la configuración no deformada.

$$\mathbf{R}_{p} = \{-\mathbf{A2^{*}L_{1}} + \mathbf{L}_{2}^{*}(\mathbf{A2^{*}A4} - \mathbf{4^{*}A1^{*}A3}), \mathbf{2^{*}(A1^{*}L_{1} - \mathbf{L}_{2}^{*}(\mathbf{A2^{*}A3} + \mathbf{A1^{*}A4})), - (\mathbf{L}_{3} + \mathbf{I}_{3v})\}; \quad (3.97)$$

Donde:

A1 = $P_0^*P_3$; A2 = 1 - 2* P_0^2 ; A3 = $Q_0^*Q_3$; A4 = 1 - 2* Q_0^2 ; (3.98)

3.4.4. Configuración deformada.

Ahora se modelará la configuración deformada, iniciando las rotaciones en el elemento terminal y finalizando en el primer elemento, ya que es la secuencia de modelación más sencilla.

Secuencia de Rotaciones y Traslaciones:

Rotación 1.- Rota el eslabón 4 con el Quaternión r4, partiendo de una configuración no deformada arbitraria (Figura 3.35).



Figura 3.35. Rotación del cuarto eslabón en la configuración deformada.

Traslación 1.- Se trasladan los eslabones 3 y 4 en la dirección e, una distancia L, . (Figura 3.36).



Figura 3.36. Traslación de los eslabones 3 y 4 en la configuración deformada.

Rotación 2.- Rotan los eslabones 2, 3 y 4 con el Quaternión r2. (Figura 3.37).



Figura 3.37. Rotación de los eslabones 2, 3 y 4 en la configuración deformada.

Rotación 3.- Rotan los cuatro eslabones con el Quaternión r1. (Figura 3.38).



Figura 3.38. Rotación de los cuatro eslabones en la configuración deformada.

De la Figura 3.35 podemos ver que los ejes de rotación de los eslabones se conservan como en la configuración original (Figura 3.30). Por lo anterior podemos definir los Quaterniones para rotar a posición deformada como se muestra a continuación.

Eslabón 1: $r1 = \{r1_0, r1_1, r1_2, r1_3\} \implies r1_v = \{r1_1, r1_2, r1_3\}, ||r1|| = 1.$

La forma local del Quaternión r1 es:

MODELACION CINEMATICA

ESTA TESIS NO DEBE

SALIR DE LA BIBLIOTECA

 $r1_{v} = Sin(\theta_{1}/2)^{*} e_{3} = Sin(\theta_{1}/2)^{*}\{0, 0, 1\}$ $r1_{v} = \{0, 0, r1_{3}\}$

De lo anterior: r1 = { r10, 0, 0, r13 }, Donde:

$$r1_0 = Cos(\theta_1/2);$$

 $r1_3 = Sen(\theta_1/2);$

Siendo 0, el ángulo de rotación del eslabón 1 en la configuración deformada.

Eslabón 2: $r2 = \{r2_0, r2_1, r2_2, r2_3\} \implies r2_v = \{r2_1, r2_2, r2_3\}, r2_v = 1.$

La forma local del Quaternión r2 es:

$$\begin{split} r2_v &= \text{Sin}(\theta_2/2)^* \ e_3 = \text{Sin}(\theta_2/2)^* \{0, \ 0, \ 1\} \\ r2_v &= \{ \ 0, \ 0, \ r2_3 \ \} \end{split}$$

De lo anterior: r2 = { r20, 0, 0, r23}, Donde:

 $r_{2_0} = Cos(\theta_2/2);$ $r_{2_3} = Sen(\theta_2/2);$

Siendo 02 el ángulo de rotación del eslabón 2 en la configuración deformada.

Eslabón 3: Como el eslabón 3 tiene una junta prismática y no rotacional como los demás, no hay Quaternión asociado a este eslabón, pero en su lugar existe una variable de longitud que caracteriza su traslación, que es L_{3x}.

Eslabón 4 (pinza): $r4 = \{r4_0, r4_1, r4_2, r4_3\} \implies r4_v = \{r4_1, r4_2, r4_3\}, ||r4|| = 1.$

La forma local del Quaternión r4 es:

 $r4_{v} = Sin(\theta_{4}/2)^{*} e_{3} = Sin(\theta_{4}/2)^{*}\{0, 0, 1\}$ $r4_{v} = \{0, 0, r4_{3}\}$

De lo anterior: r4 = { r40, 0, 0, r43 }, Donde:

 $r4_0 = Cos(\theta_4/2);$ $r4_3 = Sen(\theta_4/2);$

Siendo 04 el ángulo de rotación de la pinza (Eslabón 4) en la configuración deformada.

Aplicación de la Secuencia de rotaciones y traslaciones para determinar la configuración deformada.

Rotación 1.- Rotación del cuarto eslabón con el Quaternión r4. Esta rotación solamente afecta al eslabón 4. (Figura 3.35).

$$\mathbf{h}_{j}^{III} = \rho(\mathbf{r4}, \mathbf{h}_{j}^{II})$$
 (Rotación del cuarto eslabón con Quaternión r4)
Donde: $\mathbf{j} = 1,2,3$ (3.99)

Traslación 1.- Se trasladan, una distancia L_{3x} , los eslabone 3 y 4 sobre el eje e_3 . Esta traslación no afecta la forma de ninguna base local. (Figura 3.36).

Rotación 2.- Rotación de los eslabones 2, 3 y 4 con el Quaternión r2. Esta rotación afecta las bases f_i , g_i y h_i . (Figura 3.37).

$f_{j}^{111} = \rho(r2, f_{j}^{11})$	(Rotación del segundo eslabón con Quaternión r2)	
$\mathbf{g}_{j}^{\mathrm{HL}} = \rho(\mathbf{r2}, \mathbf{g}_{j}^{\mathrm{H}})$	(Rotación del tercer eslabón con Quaternión r2)	(3.100)
$h_j^{IV} = \rho(r2, h_j^{III})$ Donde: j = 1,2,3	(Rotación del cuarto eslabón con Quaternión r2)	

Rotación 3.- Rotación de los cuatro eslabones con el Quaternión r1. Esta rotación, que es la última, afecta a todas las bases.. (Figura 3.38).

$e_{j}^{11} = \rho(r1, e_{j}^{1})$	(Rotación del primer eslabón con Quaternión r1)	
$\mathbf{f}_{j}^{\mathrm{IV}} = \rho(\mathbf{r1}, \ \mathbf{f}_{j}^{\mathrm{III}})$	(Rotación del segundo eslabón con Quaternión r1)	
$\mathbf{g}_{j}^{\mathrm{IV}} = \rho(\mathbf{r1}, \mathbf{g}_{j}^{\mathrm{III}})$	(Rotación del tercer eslabón con Quaternión r1)	(3.101)
$\mathbf{h}_{\mathbf{j}}^{\mathbf{V}} = \rho(\mathbf{r1}, \mathbf{h}_{\mathbf{j}}^{\mathbf{IV}})$	(Rotación del cuarto eslabón con Quaternión r1)	
Donde: j = 1,2,3		

La forma final resultante de las bases locales de cada eslabón es la siguiente:

Eslabón 1.-

$$\mathbf{e}_{j}^{II} = \rho(\mathbf{r1}, \rho(\mathbf{P}, \mathbf{e}_{j})) = \rho(\mathbf{r1}^{*}\mathbf{P}, \mathbf{e}_{j})$$
 (3.102)

Eslabón 2.-

$$f_{j}^{IV} = \rho(r1, f_{j}^{III}) = \rho(r1, \rho(r2, f_{j}^{II})) = \rho(r1, \rho(r2, \rho((P^{*}Q), f_{j}))) = \rho(r1^{*}r2^{*}P^{*}Q, f_{j})$$

$$f_{j}^{IV} = \rho(r1, \rho(r2, \rho((P^{*}Q), e_{j}))) = \rho(r1^{*}r2^{*}P^{*}Q, e_{j})$$
(3.103)

Eslabón 3.-

$$g_{j}^{(V)} = \rho(r1, g_{j}^{(II)}) = \rho(r1, \rho(r2, g_{j}^{(II)})) = \rho(r1, \rho(r2, \rho((P^{*}Q), g_{j})))$$

$$g_{j}^{(V)} = \rho(r1, \rho(r2, \rho((P^{*}Q), e_{j}))) = \rho(r1^{*}r2^{*}P^{*}Q, e_{j})$$
(3.104)

Eslabón 4.-

$$\mathbf{h}_{j}^{V} = \rho(\mathbf{r1}, \mathbf{h}_{j}^{W}) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \mathbf{h}_{j}^{H})) = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{r4}, \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{h}_{j})))$$

$$\mathbf{h}_{j}^{V} = \rho(\mathbf{r1}, \rho(\mathbf{r2}, \rho(\mathbf{r4}, \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{e}_{j})))) = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r4}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{e}_{j})$$

$$(3.105)$$

3.4.5. Vectores que definen los eslabones y posición del elemento terminal en la configuración deformada.

De la Figura 3.38 y Figura 3.39 se determinan los vectores que definen los eslabones del manipulador y el vector del posición del elemento terminal (punto p), que en este caso, se considera ser un punto intermedio de la pinza, definido por el vector b_4 . Siendo L_1 , L_2 , L_{3v} y L_4 las longitudes de los eslabones y I_{3v} la posición del eslabón 3 en la posición no deformada.

b ₁ = L ₁ * e ¹ ₁ ;	(Eslabón 1)	
$b_2 = L_2^* f_1^{111};$	(Eslabón 2)	
$b_{3v} = L_{3}^{*}(-g_{3}^{V});$	(Eslabón 3 fijo)	
$b_{3v} = (L_{3v} + I_{3v})^* (-g_3^V);$	(Eslabón 3 variable)	(3.106)
$b_4 = L_4^*(-h_3^{VI});$	(Pinza)	
b _s = L _s *(- h ₁ ^{V1});	(Pinza)	

$$\mathbf{r}_{\mathbf{p}} = \mathbf{b}_1 + \mathbf{b}_2 + \mathbf{b}_3 + \mathbf{b}_{3v} + \mathbf{b}_4 \quad (\text{Posición del elemento terminal}) \quad (3.107)$$



Figura 3.39. Vector de posición del elemento terminal en la configuración deformada.

3.4.6. Ecuaciones explícitas optimizadas para la configuración deformada.

Las expresiones anteriores se explicitaron y optimizaron utilizando Mathematica®, y se presentan a continuación.

$$b_{1} = \{B1^{*}L_{1}, -2^{*}B2^{*}L_{1}, 0\}$$

$$b_{2} = \{C1^{*}L_{2}, -2^{*}C2^{*}L_{2}, 0\}$$

$$b_{3} = \{0, 0, -L_{3}\}$$

$$b_{3v} = \{0, 0, -(L_{3v}+I_{3v})\}$$

$$b_{4} = \{0, 0, -L_{4}\}$$

$$b_{5} = \{(4^{*}A9^{*}C2 - A10^{*}C1)^{*}L_{5}, 2^{*}(A9^{*}C1 + A10^{*}C2)^{*}L_{5}, 0\}$$
(3.108)

$$\mathbf{r}_{p} = \{\mathbf{B1^{*}L_{1}} + \mathbf{C1^{*}L_{2}}, -\mathbf{2^{*}(B2^{*}L_{1}} + \mathbf{C2^{*}L_{2}}), - (\mathbf{L}_{3} + \mathbf{L}_{3}, + \mathbf{L}_{4})\}$$
(3.109)

Donde:

A1 = P ₀ *P ₃ ; A2 = 1-2*	P ₀ ² ;	
A3 = Q. Q.; A4 = 1-2"	Q ₀ ² ;	
A5 = r1,"r1; A6 = 1-2"	r1 ₀ 2;	
A7 = r2, r2; A8 = 1-2*	r2,°;	(3.110)
A9 = r40"r4; A10 = 1-2"	r4 ₀ ² ;	
B1 = A2*A8 - 4*A1*A7;	B2 = A2*A7 + A1*A8;	
B3 = A10*A8 - 4*A7*A9;	B4 = A10*A7 + A8*A9;	
B5 = A2*A4 - 4*A1*A3;	B6 = A2*A3 + A1*A4;	
C1 = B4*B5 + B3*B6;	C2 = 83*85 - 4*84*86;	

CAPITULO IV

4. SIMULACION COMPUTACIONAL DE LA CINEMATICA.

La cinemática de un robot consiste en describir la relación entre el movimiento de las juntas del manipulador y el movimiento resultante de los cuerpos rígidos que constituyen el manipulador robótico. En este capítulo presentaremos la simulación computacional de la cinemática directa y cinemática inversa de los manipuladores modelados en el capítulo anterior, dicha simulación se realizará utilizando el lenguaje de cálculo formal Mathematica[®], que hemos venido utilizando en el desarrollo de este trabajo. La cinemática directa de un robot consiste en determinar la posición, y orientación en su caso, del elemento terminal (pinza o herramienta montada en el extremo del robot) cuando se dan las configuraciones relativas de cade par de eslabones adyacentes del robot. En este trabajo solo trataremos manipuladores de cadena abierta, en los cuales los eslabones forman una cadena simple y cada par de eslabones esta conectado por medio de una junta rotacional o prismática. El problema de la cinemática inversa consiste en que dada la posición, y orientación si es el caso, del elemento terminal, se deben determinar los ángulos que deberán desplazarse las diferentes juntas del manipulador para alcanzar la posición y orientación deseada; lo anterior se realiza planteando un sistema de ecuaciones donde se involucran los parámetros de rotación de cada eslabón, así como también, la posición y orientación del elemento terminal.

4.1 Cinemática de un robot de tres grados de libertad.

En la siguiente figura se muestra la configuración del robot que se simulará en está sección, el cual fue prevlamente modelado en el capítulo anterior.



Figura 4.1. Configuración del manipulador de tres grados de libertad

4.1.1 Cinemática Directa.

El problema cinemático directo, en este caso, consiste en que dados los desplazamientos angulares θ_1 , θ_2 y θ_3 , se debe determinar la posición del elemento terminal, así como la configuración resultante de sus eslabones. Para realizar lo anterior hacemos uso de las ecuaciones (3.41), (3.42), (3.43) determinadas en el capitulo anterior, que al evaluarlas se resuelve la cinemática directa de este manipulador. Dichas ecuaciones se muestran a continuación.

$$b_1 = \{0, 0, L_1\}$$

$$b_2 = \{-4^*A5^*B2^*L_2, -2^*A6^*B2^*L_2, B1^*L_2\}$$

$$b_3 = \{-4^*A5^*C1^*L_3, -2^*A6^*C1^*L_3, C2^*L_3\}$$
(4.1)

$$\mathbf{r}_{s} = \{-4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}L_{3}), -2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}L_{3}), L_{1} + B1^{*}L_{2} + C2^{*}L_{3}\}$$
(4.2)

Siendo:

A1 =
$$P_0^*P_1$$
; A2 = 1-2* P_0^2 ;
A3 = $Q_0^*Q_1$; A4 = 1-2* Q_0^2 ;
A5 = $r1_0^*r1_3$; A6 = 1-2* $r1_0^2$;
A7 = $r2_0^*r2_1$; A8 = 1-2* $r2_0^2$;
A9 = $r3_0^*r3_1$; A10 = 1-2* $r3_0^2$; (4.3)
B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8;
B3 = A10*A8 - 4*A1*A9; B4 = A10*A7 + A8*A9;
B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4;
C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6;

Las variables de las ecuaciones anteriores tienen el siguiente significado:

 $\begin{array}{l} \mathsf{L}_1, \mathsf{L}_2, \mathsf{L}_3 \rightarrow \mathsf{Longitud} \ \text{de los eslabones del manipulador.} \\ \mathsf{b}_1, \mathsf{b}_2, \mathsf{b}_3 \rightarrow \mathsf{Eslabones del manipulador.} \\ \mathsf{r}_p \rightarrow \mathsf{Posición} \ \text{del elemento terminal.} \\ \mathsf{P}_0, \mathsf{P}_1, \mathsf{Q}_0, \mathsf{Q}_1 \rightarrow \mathsf{Parámetros} \ \text{de rotación en la configuración no deformada.} \\ \mathsf{r1}_0, \mathsf{r1}_3, \mathsf{r2}_0, \mathsf{r2}_1, \mathsf{r3}_0, \mathsf{r3}_1 \rightarrow \mathsf{Parámetros} \ \text{de rotación en la configuración deformada.} \end{array}$

Los parámetros de rotación tienen la siguiente relación con las rotaciones de los eslabones del manipulador.

Siendo:

φ₁ el ángulo de rotación del eslabón 2 en la configuración no deformada.
 φ₂ el ángulo de rotación del eslabón 3 en la configuración no deformada.

 θ_1 el ángulo de rotación del eslabón 1 en la configuración deformada. θ_2 el ángulo de rotación del eslabón 2 en la configuración deformada.

0, el ángulo de rotación del eslabón 3 en la configuración deformada.

Antes de poder realizar el proceso de cinemática directa del manipulador, debemos determinar la configuración no deformada del robot mediante un proceso de cinemática inversa, que consiste en calcular los parámetros $P=\{P_0,P_1,0,0\}$ y $Q=\{Q_0,Q_1,0,0\}$ dada la posición del elemento terminal R_p (Ecuaciones (3.11), (3.12)), que es el extremo libre del eslabón 3. Es decir, con las siguientes ecuaciones puede plantearse un sistema de ecuaciones, como se muestra.

$$\mathbf{R}_{p} = \{\mathbf{0}, \ \mathbf{2}^{*}((\mathbf{A}\mathbf{2}^{*}\mathbf{A}\mathbf{3} + \mathbf{A}\mathbf{1}^{*}\mathbf{A}\mathbf{4}) \ \mathbf{L}_{3} \cdot \mathbf{A}\mathbf{1}^{*}\mathbf{L}_{2}\}, \ \mathbf{L}_{1} - \mathbf{A}\mathbf{2}^{*}\mathbf{L}_{2} + (\mathbf{A}\mathbf{2}^{*}\mathbf{A}\mathbf{4} - \mathbf{4}^{*}\mathbf{A}\mathbf{1}^{*}\mathbf{A}\mathbf{3})^{*}\mathbf{L}_{3}\}; \quad (4.6)$$

Donde:

A1 =
$$P_0^*P_1$$
; A2 = 1 - 2* P_0^2 ; (4.7)
A3 = $Q_0^*Q_1$; A4 = 1 - 2* Q_0^2 ;

Con la ecuación para R_p podemos plantear 2 ecuaciones si damos la posición para el elemento terminal como $R_p = \{0, R_py, R_pz\}$ y con las normas unitarias de los Quaterniones P y Q completamos un sistema de 4 ecuaciones con 4 incógnitas, que al resolverlo obtenemos los parámetros que determinan la configuración no deformada.

$$2^{*}((A2^{*}A3 + A1^{*}A4) *L_{3} - A1^{*}L_{2}) = R_{p}y$$

$$L_{1} - A2^{*}L_{2} + (A2^{*}A4 - 4^{*}A1^{*}A3)*L_{3} = R_{p}z$$

$$P_{0}^{2} + P_{1}^{2} = 1$$

$$Q_{0}^{2} + Q_{1}^{2} = 1$$
(4.8)

Con lo anterior podemos ahora resolver el problema cinemático directo y realizar la correspondiente simulación computacional, que como se mencionó se realizará en Mathematica[®]. A continuación se presenta el programa desarrollado para este caso, que se mostrará en dos columnas, a la derecha el programa propiamente dicho y a la izquierda una breve explicación del programa. Con el objeto de mostrar la forma de utilizar Mathematica[®] para obtener las ecuaciones del manipulador, tanto en su configuración no deformada como en la deformada, se incluye el proceso completo para su obtención en esta parte del programa.

En la primera parte del programa se define la transformación rotación de un vector	Definición de la Transformación Rotación.
tridimensional y de un Quaternión, y de algunas otras funciones necesarias que se	(* Se apagan algunos mensajes del sistema *)
muestran a continuación.	Off[General::spell1] Off[General::spell]
Transformación de un Quaternión en un vector tridimensional.	(* Se define la Transformación de un Quaternión a un vector tridimensional Tv[] *)
$q = \{q_0, q_1, q_2, q_3\} \in Q$	Tv[q_]:={q[[2]],q[[3]],q[[4]]};
Tv[q] = {q ₁ , q ₂ , q ₃ }.	(* Se define la Transformación inversa de Tv[] *)
Transformación vectorial inversa.	Tvinv[v_]:={0.v[[1]].v[[2]].v[[3]]};

$v = \{v_1, v_2, v_3\} \in \Re^3$	(* Se define la Transformación de un Quaternión a Quaternión vectorial *)
$Tvinv[v] = \{0, v_1, v_2, v_3\}.$	TvQ[q_]:={0,q[[2]],q[[3]],q[[4]]};
Transformación de un Quaternión en un Quaternión vectorial.	(* Se define la Multiplicación entre dos Quaterniones *)
$TvQ[q] = \{0,q_1, q_2, q_3\}.$	ProductoQ[p_,q_]:=
Producto de dos Quaterniones.	$p[[1]]^{a}[[2]] + p[[2]]^{a}[[1]] + p[[3]]^{a}[[4]] + p[[4]]^{a}[[3]],$ $p[[1]]^{a}[[3]] - p[[2]]^{a}[[4]] + p[[3]]^{a}[[1]] + p[[4]]^{a}[[2]],$
$p = \{p_0, p_1, p_2, p_3\}, q = \{q_0, q_1, q_2, q_3\} \in Q$	p[[1]]*q[[4]] + p[[2]]*q[[3]] - p[[3]]*q[[2]] + p[[4]]*q[[1]]};
ProductoQ[p,q] =	(* Se define el conjugado de un Quaternión *)
$(p_0^*q_0 \cdot p_1^*q_1 \cdot p_2^*q_2 \cdot p_3^*q_3, p_0^*q_1 + p_1^*q_0 + p_2^*q_3 \cdot p_3^*q_2,$	ConjugadoQ[p_]:={p[[1]],-p[[2]],-p[[3]],-p[[4]]);
Po [*] Q ₂ + P1 [*] Q ₂ + P2 [*] Q ₀ + P3 [*] Q ₁ , Po [*] Q ₃ + P1 [*] Q ₂ - P2 [*] Q ₁ + P3 [*] Q ₀)	(* Se define la Transformación Rotación de un vector tridimensional *)
Conjugado de un Quaternión.	Rotacionin y l'=
ConjugadoQ[p] = { $p_0, -p_1, -p_2, -p_3$ }	Tv[ProductoQ[p, ProductoQ[TvInv[v], ConjugadoQ[p]]]];
Rotación de un vector v e R3 con un	
Quaternión p, siendo p = 1 . Según la definición (2.13).	(* Se define la Transformación Rotación de un Quaternión *)
$\rho(\mathbf{p},\mathbf{q}) = \frac{1}{\left\ \mathbf{p}\right\ ^2} \bullet \left(\mathbf{p} \bullet \mathbf{q} \bullet \overline{\mathbf{p}}\right), \forall \mathbf{q} \in \mathbf{Q}$	RotacionQ[p_,q_]:={q[[1]],0,0,0}+ TvQ[ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]];
Rotacion[p,v] = Tv[p * Tvinv[v] * p]	
Rotación de q e Q con un Quaternión p.	
siendo p = 1. De acuerdo con la definición	
(2.13) y considerando que el primer elemento	
de q no cambia al rotar cuando p = 1	
RotacionQ[p,q] = {q ₀ ,0,0,0} + TvQ[p*q* p]	
Se realizan definiciones iniciales necesarias,	Definiciones iniciales
como son la base canónica de 33 y las bases	(* Base canónica de R3*)
eslabones. También se define la forma local	$e1 = \{1,0,0\}; e2 = \{0,1,0\}; e3 = \{0,0,1\};$
de los Quaterniones que nos permitirán rotar,	(* Bases locales de los eslabones *)
deformada.	(* Eslabón 1 *)
	e1 = {1,0,0}; e2 = {0,1,0}; e3 = {0,0,1};
	(* Eslabón 2 *) f1 = (1 0 0); f2 = (0 1 0); f3 = (0 0 1);
	1 = 1, 0, 0, 0, 0 = 10, 0, 0 = 10, 0, 0, 0

	(* Eslabón 3 *)
	$g1 = \{1,0,0\}; g2 = \{0,1,0\}; g3 = \{0,0,1\};$
	(*Quaterniones para rotar a posición no deformada*
	P = {P0,P1,0,0}; (* Eslabón 2 *) Q = {Q0,Q1,0,0}; (* Eslabón 3 *)
	(* Quaterniones para rotar a posición deformada *)
	r1 = {r10, 0, 0, r13}; (* Eslabón 1 *) r2 = {r20, r21, 0, 0}; (* Eslabón 2 *) r3 = {r30, r31, 0, 0}; (* Eslabón 3 *)
En esta sección del programa se realiza la modelación de la configuración no deformada del manipulador. Esta se obtigna rotando con	MODELADO DE LA CONFIGURACION NO DEFORMADA
la secuencia 3-2 (primero se rota el tercer eslabón y después el segundo).	Modelo para la configuración no deformada (Secuencia 3-2)
Se definen las normas unitarias de los Quaterniones P y Q y se almacenan en las variables "normas1" y "normas2", para	(* Se inicializan variables *) ClearAll[P0,P1,Q0,Q1,L1,L2,L3,I1,I2,I3];
aplicarlas en las ecuaciones resultantes de las rotaciones.	(* Se definen variables para las normas unitarias de los Quaterniones P y Q *) normas1 = {P1->(1-P0^2)^(1/2), Q1->(1-0^2)^(1/2)};
Las variables que se utilizan para las bases locales de los eslabones son:	normas2 = ${(1-P0^2)^{(1/2)}-P1, (1-Q0^2)^{(1/2)}-Q1};$
ej = Primer estabón.	(* Rotación de bases locales y ejes de rotación. Se rota primero el tercer eslabón y luego el segundo *)
fj = Segundo eslabón. gj = Tercer eslabón. j = 1,2,3.	(* Rotación del tercer eslabón (base "g") con Quaternión Q *)
Para identificar las bases después de aplicar rotaciones, se les agrega a cada variable un número que identifica la rotación aplicada.	g11 = Rotacion[Q.g1]//.normas1//.normas2; g21 = Rotacion[Q.g2]//.normas1//.normas2; g31 = Rotacion[Q.g3]//.normas1]//.normas2;
	(* Rotación del segundo eslabón (base "f") con Quaternión P *)
	f11 = Rotacion[P,f1]//.normas1//.normas2; f21 = Rotacion[P,f2]//.normas1//.normas2; f31 = Rotacion[P,f3]//.normas1//.normas2;
	(* Rotación del tercer eslabón (base "g") con Quaternión P *)
	g12 = Expand[Expand[Rotacion[P,g11]] //.normas1]//.normas2; g22 = Expand[Expand[Rotacion[P,g21]] //.normas1]//.normas2; g32 = Expand[Expand[Rotacion[P,g31]]

del elemento terminal son identificados, en la posición no deformada, por las variables siguientes: manipulador *) b1 = Esiabón 1. b2 = Esiabón 2. b3 = Esiabón 3. b1 = L1*e3; (* Esiabón 1 *) b2 = L2*31; (* Esiabón 2 *) b3 = L3*g32; (* Esiabón 3 *) Rp = b1+b2+b3; constantes que se repiten y que servirán para simplificar las ecuaciones y optimizar operaciones *) Rp = b1+b2+b3; constantes (P0*P1->A1, P0*2->(1-A2)/2, Q0*Q1->A3, Q0*Q->(1-A4)/2); Las ecuaciones se simplifican utilizando las siguientes constantes. b2 = b2// constantes; b3 = Expand[Expand[b3// constantes]// constantes]; A1 = P0*P1; A2 = 1 - 2*P0*2; A3 = Q0*Q1; A4 = 1 - 2*Q0*2; (*Vector que define posición del elemento terminal *) A3 = Q0*Q1; A4 = 1 - 2*Q0*2; Es esta parte se muestran las ecuciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no coderonada del a cinematica directa del manipulador en cuestión. Es decir, b1 = (0, 0, L1); (* Esiabón 1 *) computacional de la cinematica directa del b1 = (0, 2*(A2*A3 + A1*A4)*L3, (* Esiabón 2 *) computacional de la cinematica directa del b2 = (0, 2*(A2*A3 + A1*A4)*L3, (* Esiabón 2 *) manipulador de tres grados de libertad, debermos determinar, la configuración no deformada realizando un proceso de cinematica inversa para calcular P y Q para una posición concoids del elemento terminal, la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Re resuelve el sistema de ecuaciones vitizan e la posición del elemento terminal, la cual debe tener la forma siguiente: Cinematica inversa para determinar la configuración no deformada realizando un proceso de	Los eslabones del manipulador y la posicio	on (* Vectores que definen los eslabones del
posición no deformada, por las variables siguientes: b1 = Esiabón 1. b2 = Esiabón 2. b3 = Esiabón 3. Rp = b1+b2+b3. Rp = b1+b2+b3. Rp = posición del elemento terminal. Las ecuaciones se simplifican utilizando las siguientes constantes. A1 = P0*P1; A2 = 1 - 2*P0*2; A3 = Q0*Q1; A4 = 1 - 2*Q0*2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición del la cinemática directa del manipulador de tras grados de libertad deformada de la configuración no deformada de la configuración no deformada en una posición del elemento terminal; antes de generar una simulación deformada de la configuración no deformada de la configuración no deformada ne valta porsector no deformada. Rp = (0,Rpy, Rpz). Rp = (0,Rpy, Rpz). Rp = (0,Rpy, Rpz). Se resuelve el sistema de ecuaciones de las encle lemento terminal; Rp = (0,Rpy, Rpz). Rp = (0,Rpy, Rpz). Se resuelve el sistema de ecuaciones de las encle las non poceso de constantes: Rp = (0,Rpy, Rpz). Rp = (0,Rpy, Rpz). Se resuelve el sistema de ecuaciones de tormata en en (4.6), para P0,P1,Q0,Q1; utilizando la función FindRoot, cuya sintaxis a puede consulta en el Apéndico, al igual que todas las funciones que se utilizantes se resuelva el el sistema de ecuaciones se utilizanto se redondean si constantes en (4.6), para P0,P1,Q0,Q1; utilizando la función FindRoot, cuya sintaxis a puede consulta en el Apéndico, al igual que todas las funciones que se utilizantes se resuelva el el sistema de ecuaciones se resuelva el sistema de ecuaciones se resuelva el el sistema de ecuaciones se resuelva el el sistema de ecuaciones se resuelva el sistema de se utilizantes se puede consulta en el Apéndico, al igual que todas las funciones que se utilizan	del elemento terminal son identificados, en	la manipulador *)
siguientes: b1 = Esiabón 1. b2 = Esiabón 2. b3 = Esiabón 3. Rp = b1+b2+b3. Rp = b1+b2+b3. Rp = posición del elemento terminal. Las ecuaciones se simplifican utilizando las siguientes constantes. A1 = P0*P1: A2 = 1 - 2*P0*2: A3 = Q0*Q1: A4 = 1 - 2*Q0*2; Re esta parte se muestran las ecuaciones se tulizarán para determinar los parámetros P y Q, que definen las configuración no deformade en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación no deformade realizando un proceso de cinemática inverse para calculer P y Q para la cuel debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Se resuelve el sistema de ecuaciones resuelve el sistema de ecuaciones se puede consulta en el Apéndice, al igual de consulta en el Apéndice, al igual de tormada las funciones que se utilizan el se puede consulta en el Apéndice, al igual de toras las funciones que se utilizan el se puede consulta en el Apéndice, al igual de toras las funciones que se utilizan el se puede consulta en el Apéndice, al igual tos resultados obtenidos se redondera si se puede consulta en el Apéndice, al igual tos resultados obtenidos se redondera si se puede consulta en el Apéndice, al igual tos resultados obtenidos se redondera si se puede consulta en el Apéndice, al igual tos resultados obtenidos se redondera si se puede consulta en el Apéndice, al igual tos resultados obtenidos se redondera si se puede consulta en el Apéndice, al igual tos resultados obtenidos se redondera si se puede consulta en el Apéndice, al igual termatica inverse para determinar la consepondientes ") Cansentes de las ecuaciones tormada terminan los Quaterniones t"P" y "Q" correspondientes ")	posición no deformada, por las variabl	25
b1 = Esiabón 1.b2 = Esiabón 2.b3 = Esiabón 3.b3 = Esiabón 3.Rp = b1+b2+b3;Rp = b1+b2+b3;Rp = Posición del elemento terminal.Las ecuaciones se simplifican utilizando lassiguientes constantes.A1 = P0*P1; A2 = 1 - 2*P0*2;A3 = Q0*Q1; A4 = 1 - 2*Q0*2;Cas esuaciones an la sección anterior, las cualesEs esta parte se muestran las ecuacionesb utilizarán para determinar los parámetrosP y Q, que definen la configuración nodeformada en una posición dada deldirecta del manipulador en cuestión. Es decir.antes de generar una simulación de la cinemáticadirecta del manipulador en cuestión. Es decir.antes de generar una simulación de la cinemáticadebemos determinar la configuración nodeformada realizando un proceso de(* Constantes del se cuaciones del elemento terminal.la cual debe tener la forma siguiente:Rp = {0,Rpy, Rpz}.Rp = {0,Rpy, Rpz}.Cas resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1;Chandtes de la secuaciones mostrado en (4.6), para P0,P1,Q0,Q1;Cas resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1;Chandtes de la secuaciones mostrado en (4.6), para P0,P1,Q0,Q1;Constantes de la secuaciones mostrado en (4.6), para P0,P1,Q0,Q1;Conciendo la posición concida del elemento eterminal;la cual debe tener la forma siguiente:Rp = {0,Rpy, Rp2}.Constantes de la secuaciones mostrado en (4.6), para P0,P1,Q0,Q1;L1 = 6; L2 = 5; L3 = 4;<	siguientes:	b1 = L1*e3; (* Eslabón 1 *)
b1 = Eslabón 1. b2 = Eslabón 3.b3 = Eslabón 3.b3 = Eslabón 3.b3 = Eslabón 3.Rp = b1+b2+b3;(* Constantes que se repiten y que servirán para simplificar las ecuaciones y optimizar operaciones *) Q0*Q1->A3, Q0*2->(1-A4)/2);Las ecuaciones es simplifican utilizando las siguientes constantes.b2 = b2// constantes; b2 = b2// constantes; b3 = Expand[Expand[b3//.constantes]//.constantes];A1 = P0*P1;A2 = 1 - 2*P0*2; A3 = Q0*Q1;A4 = 1 - 2*Q0*2;A3 = Q0*Q1;A4 = 1 - 2*Q0*2;(* Cetor que define posición del elemento terminal *) Rp = b1+b2+b3;Es esta parte se muestran las ecuaciones se utilizarán para determinar los parámetros p y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decido manipulador de tres grados de libertad debemos determinar la configuración no deformada realizando un proceso de cinemática inverse para calcular P y Q para una posición conocida del elemento terminal; la cual debe tener la forma siguiente:(* Constantes de las ecuaciones *) (* Posicion el elemento terminal; la cual debe tener la forma siguiente:Rp = {0,Rpy, Rpz}.(* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0*2; A3 = Q0*Q1; A4 = 1 - 2*Q0*2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4;Se resuelve el sistema de ecuaciones se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo.Cinemática Inversa para determinar la Configuración no DeformadaSe resuelve el sistema de ecuaciones se puede consulta en el Apéndice, al igual que todas las fun		b2 = L2*f31; (* Eslabón 2 *)
b2 = Esiabón 2. b3 = Esiabón 3. Rp = b1+b2+b3;(* Constantes que se repiten y que servirán para simplificar las ecuaciones y optimizar operaciones *)Rp = b1+b2+b3;constantes que se repiten y que servirán para simplificar las ecuaciones y optimizar operaciones *)Rp = Posición del elemento terminal.Q0*Q1->A3, Q0*2->(1-A4)/2);Las ecuaciones se simplifican utilizando las siguientes constantesb2 = b2// constantes;A1 = P0*P1; A3 = Q0*Q1; A3 = Q0*Q1;A4 = 1 - 2*Q0*2;Es esta parte se muestran las ecuaciones obtenidas en la sacción anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de tres grados de libertad. deformada realizando un proceso de formada realizando un proceso de inemática inverse para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente:Exelabon 2.*Rp = {0,Rpy, Rpz}.ClearAil[P0,P1,Q0,Q1] (* Posición el elemento terminal *) b2 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) manipulador de tres grados de libertad. deformada realizando un proceso de inemática inverse para calcular P y Q para una posición conocida del elemento terminal. la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.Rp = {0,2*(A2*A3 + A1*A4)*L3 - A1*L2), L1 = 6; L2 = 5; L3 = 4;Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan e estertabajo.Clenemática Inverse para determinar la Configuración no Deformada (* Conociendo la posición del el	b1 = Eslabón 1.	b3 = L3*g32; (* Eslabón 3 *)
b3 = Esiabón 3. (* Constantes que se repiten y que servián para simplificar las ecuaciones y optimizar operaciones *) Rp = Dosición del elemento terminal. Constantes que se repiten y que servián para simplificar las ecuaciones y optimizar operaciones *) Las ecuaciones se simplifican utilizando las siguientes constantes. Constantes (P0*P1->A1, P0*2->(1-A4)/2); A1 = P0*P1; A2 = 1 - 2*P0*2; A3 = Q0*Q1; A4 = 1 - 2*Q0*2; A3 = Q0*Q1; A4 = 1 - 2*Q0*2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parametros deformada en una posición dada del elemento terminal, que será de donade del elemento terminal de la configuración no deformada en una posición del a clemática directa del manipulador en cuestión. Es decir, anteputador de tres grados de libertad, deformada le la configuración no deformada un proceso de clemática inversa para calcular P y Q para una posición conocida del elemento terminal *) ClearAll[P0,P1,Q0,Q1] (* Esiabón 1 *) Rp = {0,Rpy, Rpz}. (* Posición el elemento terminal *) Rp = {0,2*(A2*A3 + A1*A4)*L3 - A1*A2)*L3; (* Esiabón 3 *) Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0*2; A3 = Q0*Q1; A4 = 1 - 2*Q0*2; Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0*2; A3 =	b2 = Eslabón 2.	
Rp = b1+b2+b3;Rp = Posición del elemento terminal.Las ecuaciones se simplifican utilizando las siguientes constantesA1 = P0*P1; A3 = Q0*Q1; A3 = Q0*Q1; A3 = Q0*Q1; A3 = Q0*Q1; A3 = Q0*Q1; A3 = Q0*Q1; A4 = 1 - 2*Q0*2;Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donda arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación toimmática inversa para calcular P y Q para una posición conocida del elemento terminal; la cual debe tener la forma siguiente:Rp = {0,Rpy, Rpz}.Rp = {0,Rpy, Rpz}.Rp = {0,Rpy, Rpz}.Rp = {0,Rpy, Rpz}.Cinemática o en (4.6), para P0,P1,Q0,Q1 utilizando la función FindRodi, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las función en edecuciones mostrado en (4.6), para P0,P1,Q0,Q1 utilizando la función FindRodi, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizar en este trabajo.Re - resuelive el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1 utilizando la función FindRodi, cuya sintaxis se puede consulta en el Apéndice, al igual consulta en el Apéndice, al igual consepondientes *)Cinemática linversa para determinar la configuración no DeformadaConsection o para P0,P1,Q0,Q1 utilizando la función FindRodi, cuya sintaxis se puede consulta en el Apéndice, al igual consepondientes *)Consection o la función FindRodi, cuya sintaxis se puede consulta en el Apéndice, al igual <b< td=""><td>b3 = Eslabón 3.</td><td>(* Constantes que se repiten y que servirán para</td></b<>	b3 = Eslabón 3.	(* Constantes que se repiten y que servirán para
Rp = b1+b2+b3; Rp = Posición del elemento terminal. Las ecuaciones se simplifican utilizando las siguientes constantes. A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; B2 = b1+b2+b3; Es esta parte se muestran las ecuaciones se utilizarán para determinar los parámetros para determinar los parámetros de generar una simulación de la cinemática directa del manipulador en cuestión. Es decir. b1 = {0, 2r(A2*A3 + A1*A4)*L3, (* Eslabón 1 *) b2 = {0, 2r(A2*A3 + A1*A4)*L3, (* Eslabón 2 *) computacional de la cinemática directa del libertad. debernos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una siguiente: (* Constantes de las ecuaciones *) Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) (* Constantes de las ecuaciones *) Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) (* Constantes de las ecuaciones *) Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) (* Constantes de las ecuaciones *) Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) (* La = 6: L2 = 5: L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1 (* Conociendo la posición del elemento terminal se determinar la configuración en de formada (* Conciendo la posición del elemento terminal *) L1 = 6: L2 = 5; L3 = 4; Cinemática Inversa para determinar la configuracion en de selabones del manipulador *)		simplificar las ecuaciones y optimizar operaciones *)
Rp = Posición del elemento terminal. constantes = {P0*P1:>A1, P0*2->(1.A2)/2, Q0*Q1>>A3, Q0*2->(1.A4)/2}; Las ecuaciones se simplifican utilizando las siguientes constantes. b2 = b2//.constantes; b3 = Expand[Expand[b3//.constantes]//.constantes]; A1 = P0*P1; A3 = Q0*Q1; A3 = Q0*Q1; A4 = 1 - 2*Q0*2; Certain anterior, las cuales eutilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dad del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación debernos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal; la cual debe tener la forma siguiente: Etuaciones optimizadas *) (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0*2; A3 = Q0*Q1; A1 = P0*P1; A2 = 1 - 2*Q0*2;	Rp = b1+b2+b3;	
Rp = Posición del elemento terminal. Q0°Q1->A3, Q0^2->(1-A4)/2); Las ecuaciones se simplifican utilizando las siguientes constantes b2 = b2//.constantes; A1 = P0°P1; A2 = 1 - 2°P0^2; A3 = Q0°Q1; A4 = 1 - 2°Q0^2; A3 = Q0°Q1; A4 = 1 - 2°Q0^2; B2 = b2//.constantes; b3 = Expand[Expand[b3//.constantes]//.constantes]; B2 = b2//.constantes; b3 = Expand[Expand[b3//.constantes]//.constantes]; B3 = Q0°Q1; A4 = 1 - 2°Q0^2; B2 = b2//.constantes; b3 = Expand[Expand[b3//.constantes]//.constantes]; B2 = b1+b2+b3; Estante se muestran las ecuaciones optimizadas pars la posición no deformada en una posición dada dei elemento terminal. B2 = b1+b2+b3; Estante de donde arranque la simulación de la cinemática directa del manipulador de tres grados de libertad. B2 = (0, 2°(A2*A3 + A1*A4)*L3, (* Esiabón 1 *) antes de generar una simulación no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal. B2 = (0, Rpy, Rpz). Rp = (0, 2°((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3]; Rp = {0, Rpy, Rpz}. K2 = P0°P1; A2 = 1 - 2*P0^2; A3 = Q0°Q1; A4 = 1 - 2*Q0^2; Rp = {0, Rpy, Rpz}. K1 = P0°P1; A2 = 1 - 2*P0^2; A3 = Q0°Q1; A4 = 1 - 2*Q0^2; Rp = {0, Rpy, Rpz}. K1 = P0°P1; A2 = 1 - 2*P0^2; A3 = Q0°Q1; A4 = 1 -		constantes = {P0*P1->A1, P0^2->(1-A2)/2,
Las ecuaciones se simplifican utilizando las siguientes constantes. A1 = P0*P1: A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del inverse para catolular P y Q para inverso para catolular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1; Material de la función siguiente: Rp = {0,Rpy, Rpz}. Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1; utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al ligual que todas las funciones que se utilizan es trabajo. Los resultados obtenidos se redondean si	Rp = Posición del elemento terminal.	Q0*Q1->A3, Q0^2->(1-A4)/2);
Las ecuaciones se simplifican utilizando las siguientes constantes. A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde rarenque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición concida del elemento terminal. la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0.P1,Q.0,Q1; KP = {0, Rpy, Rpz}. Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0.P1,Q.0,Q1; L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en q Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si		
siguientes constantes b3 = Expand[Expand[b3//.constantes]//.constantes]; A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde directa del manipulador en cuestión. Es decir, antes de generar una simulación b1 = (0, 0, L1; (* Eslabón 1 *) arranque la simulación de la cinemática directa del manipulador de tres grados de libertad. b1 = (0, 0, L1; (* Eslabón 2 *) computacional de la cinemática directa del manipulador de tres grados de libertad. b1 = (0, 0, L1; (* Eslabón 3 *) cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: (* Posición el elemento terminal, la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = A1*A1*A3)*L3]; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1; A2 = 5; L3 = 4; Cinemática inversa para determinar la configuración en detormáda Cinemática inversa para determinar la configuración en el Apéndice, al igual que todas las funciones que se utilizan en el Apéndice, al igual que todas las funciones que se utilizan en el Apéndice, al igual que todas las funciones que se utilizan en el Apéndice, al igual que todas las funciones que se utilizan en el Apéndice, al igu	Las ecuaciones se simplifican utilizando la	as b2 = b2//.constantes;
A1 = P0°P1; A2 = 1 - 2°P0°2; A3 = Q0°Q1; A4 = 1 - 2°Q0°2; Rp = b1+b2+b3; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada. Py Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación no deformada realizando un proceso de libertad. debemos determinar la configuración no deformada realizando un proceso de libertad. la cual debe tener la forma siguiente: (* Cosición el elemento terminal *) (A2°A4 - 4*A1*A3)*L3; (* Eslabón 3 *) (A2°A4 - 4*A1*A3)*L3; Rp = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0°P1; A2 = 1 - 2°P0^2; A3 = Q0°Q1; A4 = 1 - 2°Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Cinemática Inversa para determinar la configuración no deformada Se resuelve el sistema de ecuaciones mostrado en el Apénice, al igual que todas las funciones que se utilizan en est trabajo. Cinemática Inversa para determinar la configuración se determinar la configuración se determinal se determinal se determinal se indexidados obtenidos se redondean si	siguientes constantes.	b3 = Expand[Expand[b3//.constantes]//.constantes];
A1 = P0*P1; A2 = 1 - 2*P0^2; (*Vector que define posición del elemento terminal *) A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (*Vector que define posición del elemento terminal *) Rp = b1+b2+b3; Rp = b1+b2+b3; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parametros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal. la cual debe tener la forma siguiente: (* Eosición el elemento terminal *) Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Cinemática Inversa para determinar la configuración no deformada realizando en (4.6), para P0,P1,Q0,Q1; (* Conociendo la posición del elemento terminal *) Rp = {0,Rpy, Rpz}. Rp = {0,Rpy, Rpz}. Rp = {0,Rpy, Rpz}. Rp = {0,Rpy, Rpz}.		
A3 = Q0°Q1; A4 = 1 - 2°Q0°2; Rp = b1+b2+b3; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestón. Es decir bal = (0, 0, L1); (* Eslabón 1 *) ba = (0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 2 *) ba = (0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3); debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: (* Posición el elemento terminal *) (* Posición el elemento terminal *) (* Posición el elemento terminal *) (* Constantes de las ecuaciones *) Rp = {0,Rpy, Rpz}. A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0°Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática Inversa para determinar la configuración no deformada Se resuelve el sistema de ecuaciones que se utilizan en este trabajo. (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *) L1 = 6; rabajo. Cinemática inversa para determinar la configuración no deformada	A1 = P0*P1; A2 = 1 - 2*P0^2;	(*Vector que define posición del elemento terminal *)
Rp = b1+b2+b3; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada dei elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de inemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: b1 = {0, 0, L1}; (* Esiabón 1*) b2 = {0, 2*A1*L2, -A2*L2}; (* Esiabón 2*) computacional de la configuración no deformada realizando un proceso de inemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Rp = {0,2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los esiabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Se resueive el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática Inversa para determinar la Configuración no Deformada	A3 = Q0*Q1; A4 = 1 - 2*Q0^2;	······································
Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros se utilizarán para determinar los parámetros deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad. debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: Ecuaciones optimizadas para la posición no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: D1 = {0, 2*(A2*A3 + A1*A4)*L3, (* Esiabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) Rp = {0,Rpy, Rpz}. Rp = {0, 2*(A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0*2; (* Longitud de los esiabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática inversa para determinar la Configuración no Deformada (* Conciendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *)		Rp = b1+b2+b3;
Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada. Ecuaciones optimizadas para la posición no deformada. P y Q, que definen la configuración no deformade en una posición dada del elemento terminal, que será de donde directa del manipulador en cuestión. Es decir. directa del manipulador en cuestión. Es decir. directa del manipulador en cuestión. Es decir. b1 = {0, 0, L1}; (* Esiabón 1*) antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad. deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal. la cual debe tener la forma siguiente: Rp = {0, Rpy, Rpz}. (* Posición el elemento terminal *)		
obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa directa directa del manipulador de tres grados de libertad. debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: the {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3; (* Posición el elemento terminal *) (* Posición el elemento terminal *) (* Posición conocida del elemento terminal, la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Rp = {0, Rpy, Rpz}. Re = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	Es esta parte se muestran las ecuacion	Ecuaciones optimizadas para la posición no
se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde elemento terminal, que será de donde intercta del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad. debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Rp = {0,Rpy, Rpz}. Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	obtenidas en la sección anterior, las cual	es deformada.
P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad. debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: I = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; Rp = {0, Rpy, Rpz}. (* Posición el elemento terminal, la cual debe tener la forma siguiente: Rp = {0, Rpy, Rpz}. Rp = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) Rp = {0, Rpy, Rpz}. Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0, P1, Q0, Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática Inversa para determinar la configuración no Deformada Conciendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *) L1 = 6; L2 = 5; L3 = 4;	se utilizarán para determinar los parámetro	S
deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad. debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; Rp = {0, Rpy, Rpz}. (* Posición el elemento terminal, la cual debe tener la forma siguiente: Rp = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si Cinemática Inversa para determinar la correspondientes *)	P v Q, que definen la configuración i	no ClearAll/P0.P1.Q0.Q11
elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: (* Ecuaciones optimizadas *) Rp = {0,Rpy, Rpz}. (* Posición el elemento terminal, la cual debe tener la forma siguiente: (* Posición el elemento terminal *) (* Constantes de las ecuaciones *) Rp = {0,Rpy, Rpz}. A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	deformada en una posición dada o	
arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad. debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal. la cual debe tener la forma siguiente: b1 = {0. 0, L1}; (* Esiabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Esiabón 3 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Esiabón 3 *) (A2*A4 - 4*A1*A3)*L3}; Rp = {0, Rpy, Rpz}. (* Posición el elemento terminal. la cual debe tener la forma siguiente: Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; Rp = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) Rp = {0, 2*(Ca*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si Cinemática Inversa para determinar la Configuración no Deformada	elemento terminal, que será de dono	de (* Ecuaciones optimizadas *)
directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Rp = {0,Rpy, Rpz}. Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	arranque la simulación de la cinemáti	
antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad. debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal. la cual debe tener la forma siguiente: b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2*) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3*) (A2*A4 - 4*A1*A3)*L3}; Rp = {0, Cartical anterial a configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal. la cual debe tener la forma siguiente: Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; Rp = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si Cinemática Inversa para determinar la Configuración no Deformada	The second s	
computacional de la cinemàtica directa del manipulador de tres grados de libertad. debemos determinar la configuración no deformada realizando un proceso de cinemàtica inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Esiabón 3 *) (A2*A4 - 4*A1*A3)*L3}; Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), la cual debe tener la forma siguiente: Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; Rp = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; K* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática Inversa para determinar la Configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *)	directa del manipulador en cuestión. Es dec	ca ir b1 = {0, 0, L1}: (* Eslabón 1 *)
manipulador de tres grados de libertad. (A2*A4 - 4*A1*A3)*L3); debemos determinar la configuración no deformada realizando un proceso de cinemàtica inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: (* Posición el elemento terminal *) Rp = {0,Rpy, Rpz}. Rp = {0, Rpy, Rpz}. Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L1 = 6; L2 = 5; L3 = 4; Cinemática Inversa para determinar la Configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *)	directa del manipulador en cuestión. Es dec antes de generar una simulación	;a ir, b1 = {0, 0, L1}; (* Eslabón 1 *) 5n b2 = {0 -2*A1*L2, -A2*L2}; (* Eslabón 2 *)
debernos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: (* Posición el elemento terminal *) Rp = {0, Rpy, Rpz}. Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; Rp = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A3 = Q0*Q1; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática Inversa para determinar la Configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *)	directa del manipulador en cuestión. Es dec antes de generar una simulación computacional de la cinemática directa d	;a ir. b1 = {0, 0, L1}; (* Eslabón 1 *) on b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) iel b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *)
deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: (* Posición el elemento terminal *) Rp = {0, Rpy, Rpz}. Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; Rp = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L1 = 6; L2 = 5; L3 = puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática Inversa para determinar la configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *) (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q"	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa d manipulador de tres grados de liberta	za ir. b1 = {0, 0, L1}; (* Esiabón 1 *) in b2 = {0, -2*A1*L2, -A2*L2}; (* Esiabón 2 *) iel b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Esiabón 3 *) d (A2*A4 - 4*A1*A3)*L3};
cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: Rp = {0, 2°((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3); Rp = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L1 = 6; L2 = 5; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática Inversa para determinar la Configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *) (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q"	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa d manipulador de tres grados de liberta debemos determinar la configuración	 a b1 = {0, 0, L1}; (* Esiabón 1 *) b0 b2 = {0, -2*A1*L2, -A2*L2}; (* Esiabón 2 *) el b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Esiabón 3 *) d. (A2*A4 - 4*A1*A3)*L3};
Ina posición conocida del elemento terminal. la cual debe tener la forma siguiente: Rp = {0, Rpy, Rpz}. Rp = {0, Rpy, Rpz}. Rp = {0, Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L1 = 6; L2 = 5; L3 = que consulta en el Apéndice, al igual que todas las funciones que se utilizan en el Apéndice, al igual que todas las funciones que se utilizan en el ste trabajo. Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa d manipulador de tres grados de liberta debemos determinar la configuración un deformada realizando un proceso de	 a b1 = {0, 0, L1}; (* Eslabón 1 *) b0 b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) el b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) d. (A2*A4 - 4*A1*A3)*L3}; d. (* Posición el elemento terminal *)
Ia cual debe tener la forma siguiente: In - A2*L2 + (A2*A4 - 4*A1*A3)*L3); Is cual debe tener la forma siguiente: L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3); Is cual debe tener la forma siguiente: L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3); Is cual debe tener la forma siguiente: L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3); Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is cual debe tener la forma siguiente: Is constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; Is constantes de las estabones del manipulador *) L1 = 6; L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, Configuración no Deformada utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en e	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa d manipulador de tres grados de liberta debemos determinar la configuración d deformada realizando un proceso de cinemática inversa para calcular P y O pa	<pre>ir. b1 = {0, 0, L1}; (* Eslabón 1 *) b1 = {0, 0, L1}; (* Eslabón 2 *) el b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) d. (A2*A4 - 4*A1*A3)*L3}; de (* Posición el elemento terminal *) ra</pre>
Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L1 = 6; L2 = 5; L3 = 4; Cinemática Inversa para determinar la mostrado en (4.6), para P0,P1,Q0,Q1, Utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual (* Conociendo la posición del elemento terminal se que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulacia computacional de la cinemática directa o manipulador de tres grados de liberta debemos determinar la configuración de deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termin	The formula f
Rp = {0,Rpy, Rpz}. (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L1 = 6; L2 = 5; L3 = 4; Cinemática Inversa para determinar la Configuración no Deformada villizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *)	directa del manipulador en cuestión. Es dec antes de generar una simulacia computacional de la cinemática directa o manipulador de tres grados de liberta debemos determinar la configuración u deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termin la cual debe tener la forma siguiente:	The formula f
A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L1 = 6; L2 = 5; L3 = 4; Cinemática inversa para determinar la Configuración no Deformada Materia de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *)	directa del manipulador en cuestión. Es dec antes de generar una simulacia computacional de la cinemática directa o manipulador de tres grados de liberta debemos determinar la configuración u deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termin la cual debe tener la forma siguiente:	$\begin{array}{llllllllllllllllllllllllllllllllllll$
A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática Inversa para determinar la Configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *) correspondientes *)	directa del manipulador en cuestión. Es dec antes de generar una simulacia computacional de la cinemática directa o manipulador de tres grados de liberta debemos determinar la configuración d deformada realizando un proceso de cinemática inversa para calcular P y Q pa una posición conocida del elemento termina la cual debe tener la forma siguiente:	$\begin{array}{l} \begin{array}{l} \begin{array}{l} \text{a}\\ \text{b1} = \{0, 0, L1\}; & (* \ \text{Eslabon 1 }*) \\ \text{b2} = \{0, -2^*\text{A1}^*\text{L2}, -\text{A2}^*\text{L2}\}; & (* \ \text{Eslabon 2 }*) \\ \text{b3} = \{0, 2^*(\text{A2}^*\text{A3} + \text{A1}^*\text{A4})^*\text{L3}, & (* \ \text{Eslabon 3 }*) \\ \text{d.} & (\text{A2}^*\text{A4} - 4^*\text{A1}^*\text{A3})^*\text{L3}\}; \\ \begin{array}{l} \text{d}\\ $
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática Inversa para determinar la Configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *) (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q"	directa del manipulador en cuestión. Es dec antes de generar una simulación computacional de la cinemática directa o manipulador de tres grados de liberta debemos determinar la configuración i deformada realizando un proceso de cinemática inversa para calcular P y Q pa una posición conocida del elemento termina la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	$\begin{array}{llllllllllllllllllllllllllllllllllll$
Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Cinemática Inversa para determinar la Configuración no Deformada (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Cinemática Inversa para determinar la Configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *)	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa o manipulador de tres grados de liberta debemos determinar la configuración i deformada realizando un proceso de cinemática inversa para calcular P y Q pa una posición conocida del elemento termina la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	$\begin{array}{l} \begin{array}{l} \begin{array}{l} \text{a}\\ \text{b1} = \{0, 0, L1\}; & (\text{* Eslabón 1 *})\\ \text{b2} = \{0, -2^*A1^*L2, -A2^*L2\}; & (\text{* Eslabón 2 *})\\ \text{b3} = \{0, 2^*(A2^*A3 + A1^*A4)^*L3, & (\text{* Eslabón 3 *})\\ \text{d}, & (A2^*A4 - 4^*A1^*A3)^*L3\};\\ \begin{array}{l} \text{d}\\ $
 (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si Cinemática Inversa para determinar la Configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *) 	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa o manipulador de tres grados de liberta debemos determinar la configuración i deformada realizando un proceso de cinemática inversa para calcular P y Q pa una posición conocida del elemento termina la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	$\begin{array}{l} \begin{array}{l} \text{a}\\ \text{b1} = \{0, 0, L1\}; & (* \ \text{Eslabon 1 *}) \\ \text{b2} = \{0, -2^*\text{A1*L2}, -\text{A2*L2}\}; & (* \ \text{Eslabon 2 *}) \\ \text{b3} = \{0, 2^*(\text{A2*A3 + A1*A4})^*\text{L3}, & (* \ \text{Eslabon 3 *}) \\ \text{d}, & (\text{A2*A4 - 4*A1*A3})^*\text{L3}\}; \\ \begin{array}{l} \text{d}\\ $
L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa co manipulador de tres grados de liberta debemos determinar la configuración i deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termina la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	$\begin{array}{l} \text{a} \\ \text{b1} = \{0, 0, \text{L1}\}; & (* \text{ Esiabón 1 *}) \\ \text{b2} = \{0, -2^*\text{A1*L2}, -\text{A2*L2}\}; & (* \text{ Esiabón 2 *}) \\ \text{b3} = \{0, 2^*(\text{A2*A3 + A1*A4})^*\text{L3}, & (* \text{ Esiabón 3 *}) \\ \text{d}, & (\text{A2*A4 - 4*A1*A3})^*\text{L3}\}; \\ \text{d} \\ \text{(* Posición el elemento terminal *)} \\ \text{ra} \\ \text{al,} \\ \text{Rp} = \{0, 2^*((\text{A2*A3 + A1*A4})^*\text{L3 - A1*L2}), \\ \text{L1 - A2*L2 + (A2*A4 - 4*A1*A3)^*\text{L3}}; \\ \text{(* Constantes de las ecuaciones *)} \\ \text{A1 = P0*P1;} \text{A2 = 1 - 2*P0^2;} \\ \text{A3 = Q0*Q1;} \text{A4 = 1 - 2*Q0^2;} \\ \end{array}$
L1 = 6; L2 = 5; L3 = 4; Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa co manipulador de tres grados de liberta debemos determinar la configuración in deformada realizando un proceso de cinemática inversa para calcular P y Q pa una posición conocida del elemento termina la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	$\begin{array}{l} \text{a} \\ \text{b1} = \{0, 0, \text{L1}\}; & (* \text{ Eslabón 1 *}) \\ \text{b2} = \{0, -2^*\text{A1*L2}, -\text{A2*L2}\}; & (* \text{ Eslabón 2 *}) \\ \text{b3} = \{0, 2^*(\text{A2*A3 + A1*A4})^*\text{L3}, & (* \text{ Eslabón 3 *}) \\ \text{d}, & (\text{A2*A4 - 4*A1*A3})^*\text{L3}\}; \\ \text{d} \\ \text{d} \\ \text{(* Posición el elemento terminal *}) \\ \text{ra} \\ \text{al}, \\ \text{Rp} = \{0, 2^*((\text{A2*A3 + A1*A4})^*\text{L3 - A1*L2}), \\ \text{L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; \\ \text{(* Constantes de las ecuaciones *}) \\ \text{A1} = \text{P0*P1}; \text{A2} = 1 - 2^*\text{P0*2}; \\ \text{A3} = \text{Q0*Q1}; \text{A4} = 1 - 2^*\text{Q0*2}; \\ \text{(* Longitud de los eslabones del manipulador *}) \end{array}$
Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa co manipulador de tres grados de liberta debemos determinar la configuración i deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termina la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	$\begin{array}{l} \text{a}\\ \text{b1} = \{0, 0, L1\}; & (* \text{ Eslabón 1 *})\\ \text{b2} = \{0, -2^*\text{A1*L2}, -\text{A2*L2}\}; & (* \text{ Eslabón 2 *})\\ \text{b3} = \{0, 2^*(\text{A2*A3 + A1*A4})^*\text{L3}, & (* \text{ Eslabón 3 *})\\ \text{d}, & (\text{A2*A4 - 4*A1*A3})^*\text{L3}\};\\ \text{d}\\ \text{d}\\ \text{(* Posición el elemento terminal *})\\ \text{ra}\\ \text{al}, & \text{Rp} = \{0, 2^*((\text{A2*A3 + A1*A4})^*\text{L3 - A1*L2}), \\ L1 - \text{A2*L2 + }(\text{A2*A4 - 4*A1*A3})^*\text{L3}\};\\ \text{(* Constantes de las ecuaciones *})\\ & \text{A1} = \text{P0*P1}; & \text{A2} = 1 - 2^*\text{P0^2}; \\ \text{A3} = \text{Q0*Q1}; & \text{A4} = 1 - 2^*\text{Q0^2};\\ \text{(* Longitud de los eslabones del manipulador *}) \end{array}$
Se resuelve el sistema de ecuaciones mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa co manipulador de tres grados de liberta debemos determinar la configuración o deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termino la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	$\begin{array}{llllllllllllllllllllllllllllllllllll$
mostrado en (4.6), para P0,P1,Q0,Q1, utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa o manipulador de tres grados de liberta debemos determinar la configuración i deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termin la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	$\begin{array}{llllllllllllllllllllllllllllllllllll$
utilizando la función FindRoot, cuya sintaxis se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa co manipulador de tres grados de liberta debernos determinar la configuración i deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termin la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	$\begin{array}{llllllllllllllllllllllllllllllllllll$
se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa co manipulador de tres grados de liberta debernos determinar la configuración i deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termin la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	$\begin{array}{llllllllllllllllllllllllllllllllllll$
que todas las funciones que se utilizan en este trabajo. Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa co manipulador de tres grados de liberta debernos determinar la configuración i deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termin la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Se resuelve el sistema de ecuacion mostrado en (4.6), para P0,P1,Q0,Q utilizando la función FindRoot, cuva sinta	air. b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; d. (A2*A4 - 4*A1*A3)*L3}; id. (* Posición el elemento terminal *) ra . ra . ail. Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) . A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; . (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L1 = 6; L2 = 5; L3 = 4;
este trabajo. Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa co manipulador de tres grados de liberta debernos determinar la configuración i deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termina la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Se resuelve el sistema de ecuaciona mostrado en (4.6), para P0,P1,Q0,Q utilizando la función FindRoot, cuya sintas se puede consulta en el Apéndice al jou	air. b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; d. (A2*A4 - 4*A1*A3)*L3}; id. (* Posición el elemento terminal *) ra ra al, Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Cinemática Inversa para determinar la Configuración no Deformada is al (* Conociendo la posición del elemento terminal se
Los resultados obtenidos se redondean si	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa co manipulador de tres grados de liberta debernos determinar la configuración o deformada realizando un proceso o cinemática inversa para calcular P y Q pa una posición conocida del elemento termina la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Se resuelve el sistema de ecuaciona mostrado en (4.6), para P0,P1,Q0,Q utilizando la función FindRoot, cuya sintas se puede consulta en el Apéndice, al igu que todas las funciones que se utilizan de	ir. b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) d. (A2*A4 - 4*A1*A3)*L3}; id. (* Posición el elemento terminal *) ra (* Posición el elemento terminal *) ra ra al, Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; es Cinemática Inversa para determinar la 1, Configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" v "O"
	directa del manipulador en cuestión. Es dec antes de generar una simulació computacional de la cinemática directa co manipulador de tres grados de liberta debemos determinar la configuración d deformada realizando un proceso d cinemática inversa para calcular P y Q pa una posición conocida del elemento termina la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}. Se resuelve el sistema de ecuaciona mostrado en (4.6), para P0,P1,Q0,Q utilizando la función FindRoot, cuya sintas se puede consulta en el Apéndice, al igu que todas las funciones que se utilizan de	ir. b1 = {0, 0, L1}; (* Esiabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Esiabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Esiabón 3 *) d. (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) ra al, Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; Cinemática Inversa para determinar la Configuración no Deformada (* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *)

alguno de ellos está muy cercano a cero.	ClearAll[P0,P1,Q0,Q1]
correspondientes (P0, P1, Q0, Q1) para	(* Se da la posición del elemento terminal *)
poder ser utilizados posteriormente, además.	
se imprimen dichos valores.	posicion = {0, 0, 15};
	(* Se determinan "P" y "Q" utilizando los siguiente valores iniciales *)
	iP0 = .7071; iP1 = .7071; iQ0 = .866; iQ1 = .5;
	(* Se almacena la solución en la variable sol y se utiliza la función FindRoot para determinar la solución del sistema de ecuaciones *)
	sol =
	FindRoot[{Rp2 == posicion[[2]],Rp3 == posicion[[3]], P0^2+P1^2 == 1, Q0^2+Q1^2 == 1}, {P0,iP0}, {P1,iP1}, {Q0,iQ0}, {Q1,iQ1}, MaxIterations->50];
	(* Se asignan los valores de la solución a las variables correspondientes *)
	P0=P0/.sol; P1=P1/.sol; Q0=Q0/.sol; Q1=Q1/.sol;
	(* Se redondean los valores de la solución *)
	lf[Abs[P0]==1,P1=0,lf[Abs[P1]==1,P0=0]]; lf[Chop[Abs[Q0]-1]==0,Q0=1;Q1=0, lf[Chop[Abs[Q1]-1]==0,Q0=0;Q1=1]];
	(* Se imprimen los resultados *)
	Print["P0 = ",P0 ," ", "P1 = ", P1, " Q0 = ", Q0, " ", "Q1 = ", Q1]
De la misma forma que se realizó la modelación de la configuración no	MODELADO DE LA CONFIGURACION DEFORMADA
deformada, ahora se presenta el desarrollo para programar la generación de la configuración deformada, que básicamente sigue al siguiente procedimiento	Modelo para la configuración deformada (Secuencia 3-2-1)
sigue er siguiente procedimiento.	ClearAll[r10,r13,r20,r21,r30,r31,L1,L2,L3];
Se parte de la forma que adquirieron las	
bases locales de los eslabones en la	(* Posición no deformada conocida *)
las rotaciones correspondientes a los tres	P = {P0,P1,0,0}; Q = {Q0,Q1,0,0};
eslabones en la secuencia 3-2-1. También en	
este caso, se definen las normas unitarias de	("Quaterniones para rotar en secuencia ")
en las variables "normas3" y "normas4", es	(* 1 Rotación para el tercer eslabón *)
de hacer notar que en estas variables se	-2 - (-20 -21 0 0)
incluyen las normas unitarias de los	$r_3 = (r_3 0, r_3 1, 0, 0);$

Quaterniones correspondientes a la	(* 2 Rotación para segundo y tercer eslabones *)
configuración no deformada con el objeto de	· · · · · · · · · · · · · · · · · · ·
lograr la mayor simplificación de las ecuaciones resultantes.	$r2 = \{r20, r21, 0, 0\};$
	(* 3 Rotación para los tres eslabones *)
	r1 = {r10,0,0,r13};
	(* Se definen normas de los Quaterniones *)
	normas3 = Flatten[{normas1, r13->(1-r10^2)^(1/2), r21->(1-r20^2)^(1/2), r31->(1-r30^2)^(1/2)}];
	normas4 = Flatten[{normas2, (1-r10^2)^(1/2)->r13, (1-r20^2)^(1/2)->r21, (1-r30^2)^(1/2)->r31}];
Se aplica la primera rotación, que consiste en	(* Se aplica primera rotación de la secuencia ("r3") *)
rotar el tercer eslabón con el Quaternión r3.	
esto afecta solamente a la base local g _j .	(* Rotación del tercer eslabón (base "g") con Quaternión r3 *)
	g13 = Simplify[Expand[Rotacion[r3,g12]]//.normas3]
	a23 = Simplify/Expand/Rotacion/r3 a2211// normas31
	//.normas4:
	g33 = Simplify[Expand[Rotacion[r3,g32]]//.normas3]
	//.normas4;
Se aplica la segunda rotación, que consiste	("Se aplica segunda rotación de la secuencia ("r2")")
Quaternión r2 que también hace girar de la	(* Rotación del segundo estabón (base "f") con
misma manera al eslabón 3 y con ello se	Quaternión r2 *)
afecta a las bases locales \boldsymbol{f}_j y $\boldsymbol{g}_j,$ que son las	f12 = Simplify[Expand[Rotacion[r2,f11]]//.normas3] //.normas4;
actualizadas.	f22 = Simplify[Expand[Rotacion[r2,f21]]//.normas3]
	//.normas4; [22 = SimplifulExpand(Batacion)/2 (21))// pormas2)
	//.normas4;
	(* Rotación del tercer eslabón (base "g") con Quaternión r2 *)
	g14 = Simplify[Expand[Rotacion[r2,g13]]//.normas3] //.normas4;
	g24 = Simplify[Expand[Rotacion[r2,g23]]//.normas3]
	a34 = Simplify[Expand[Rotacion[r2.a33]]//.normas3]
	//.normas4;
Se aplica la tercera rotación, que consiste en	(* Se aplica tercera rotación de la secuencia ("r1") *)
rotar el primer esiabon con el Quaternion 11,	 (* Rotación del primer eslabón (base "e") con
otros dos eslabones y con ello se afecta a las	Quaternión r1 *)

bases locales e_j , f_j y g_j , que son las actualizadas. Al aplicar las rotaciones, también se simplifican las expresiones obtenidas, aplicando las normas unitarias de los Quaterniones y funciones de simplificación propias de Mathematica [§] .	<pre>e11 = Simplify[Expand[Rotacion[r1,e1]]//.normas3] //.normas4; e21 = Simplify[Expand[Rotacion[r1,e2]]//.normas3] //.normas4; e31 = Simplify[Expand[Rotacion[r1,e3]]//.normas3] //.normas4; (* Rotación del segundo eslabón (base "f") con Quaternión r1 *) f13 = Simplify[Expand[Rotacion[r1,f12]]//.normas3] //.normas4; f23 = Simplify[Expand[Rotacion[r1,f22]]//.normas3] //.normas4; f33 = Simplify[Expand[Rotacion[r1,f32]]//.normas3] //.normas4; f33 = Simplify[Expand[Rotacion[r1,f32]]//.normas3] //.normas4;</pre>
	Quaternión r1 *) g15 = Simplify[Expand[Rotacion[r1,g14]]//.normas3] //.normas4; g25 = Simplify[Expand[Rotacion[r1,g24]]//.normas3] //.normas4; g35 = Simplify[Expand[Rotacion[r1,g34]]//.normas3] //.normas4;
Una vez obtenidas las bases locales en su forma final, después de aplicadas todas las rotaciones propuestas, se obtienen las ecuaciones que representan los eslabones del manipulador y la posición del elemento terminal como:	(* Vectores que definen los eslabones del manipulador *) b1 = L1*e31; (* Eslabón 1 *) b2 = L2*f33; (* Eslabón 2 *) b3 = L3*g35; (* Eslabón 3 *)
$b_1 = L_1^* e_3^1 = L_1^* e_3^1;$ (Eslabón 1) $b_2 = L_2^* f_3^{111} = L_2^* f_3^3;$ (Eslabón 2)	(* Constantes que se repiten y que servirán para simplificar las ecuaciones y optimizar operaciones *)
$b_3 = L_3^* g_3^V = L_3^* g_3^S;$ (Eslabón 3)	constantes = {P0*P1->A1, P0^2->(1-A2)/2, Q0*Q1->A3, Q0^2->(1-A4)/2,
(Posición del elemento terminal)	r10*r13->A5, r10^2->(1-A6)/2, r20*r21->A7,r20^2->(1-A8)/2,
$\mathbf{r_p} = \mathbf{b_1} + \mathbf{b_2} + \mathbf{b_3}$	r30*r31->A9,r30^2->(1-A10)/2, A2*A8->4*A1*A7+B1, A2*A7->B2-A1*A8,
Estas expresiones son simplificadas sustituyendo por constantes a aquellos términos que se repiten dentro de las diferentes ecusciones como se puede	A10*A8->4*A7*A9+B3, A10*A7->B4-A8*A9, A2*A4->4*A1*A3+B5, A2*A3->B6-A1*A4, B4*B5->C1-B3*B6, B3*B5->C2+4*B4*B6};
observar.	(* Se simplifican las expresiones *)
	b2 = Expand[b2//.constantes]; b2 = Expand[Collect[b2,{L2,A5,A6}]//.constantes];
	b3 = Expand[b3//.constantes]; b3 = Expand[Collect[Collect[b3,{L3,A5,A1,A2,A3,A4, A6}] //.constantes,{L3,A5,A6,B3,B4}]//.constantes]; b3 = Expand[b3//.constantes];

	(*Vector que define posición del elemento terminal *)
	(· · · · · · · · · · · · · · · · · · ·
	rp = b1 + b2 + b3;
En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador	Ecuaciones optimizadas para la posición deformada (Secuencia 3-2-1)
así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que	ClearAll[r10,r13,r20,r21,r30,r31,L1,L2,L3];
propiamente se utiliza para realizar la simulación de la cinemática directa del	(* Ecuaciones Simplificadas *)
manipulador de tres grados de libertad, ya que es la que contiene las ecuaciones va	b1 = {0, 0, L1};
simplificadas. Puede observarse, que además de las ecuaciones de los eslabones	b2 = {-4*A5*B2*L2, -2*A6*B2*L2, B1*L2};
y de la posición del elemento terminal, se tiene una equación para e11 a el : ésta sinve	b3 = {-4*A5*C1*L3, -2*A6*C1*L3, C2*L3};
para mostrar el movimiento del primer	rp = {-4*A5*(B2*L2 + C1*L3), -2*A6*(B2*L2 + C1*L3),
requiere que se dibuje una de las bases	L1 + B1*L2 + C2*L3};
movimiento.	(* Componentes ortogonales de rp *)
como al realizar la simulación es necesario evaluar numéricamente las ecuaciones del	rp1 = rp[[1]]; rp2 = rp[[2]]; rp3 = rp[[3]];
dichas ecuaciones en esta misma sección, almacenándolas en nuevas variables.	(* Componente 1 de la Base local del eslabón 1 (para dibujar la base) *)
	e11 = {-A6,2*A5, 0};
	(* Constantes de las ecuaciones *)
	A1 = P0*P1; A2 = 1-2*P0^2; A3 = 00*01; A4 = 1-2*00^2;
	$A5 = r10^{\circ}r13; A6 = 1-2^{\circ}r10^{2};$
	A7 = r20*r21; A8 = 1-2*r20^2;
	A9 = r30*r31; A10 = 1-2*r30^2;
	B1 = A2*A8 - 4*A1*A7;
	B2 = A2*A7 + A1*A8;
	$B3 = A10^{\circ}A8 - 4^{\circ}A7^{\circ}A9;$
	B4 = A 0 A + A0 A0, $B5 = A2^* \Delta A = A^* \Delta 1^* \Delta 3^{-1}$
	$B6 = A2^*A3 + A1^*A4$
	C1 = B4*B5 + B3*B6;
	C2 = B3*B5 - 4*B4*B6;
	(* Longitud de los eslabones del manipulador *)
	L1 = 6; L2 = 5; L3 = 4;
	(* Respaldo de ecuaciones *)
	a11=e11; z1=b1; z2=b2; z3=b3; rx=rp;

Finalmente, en esta parte del programa, se	CINEMATICA DIRECTA
nos permitirán observar la simulación de la cinemática, directa, del manipulador, en	Graficación para visualizar la simulación.
cuestión.	ClearAll[r10,r13,r20,r21,r30,r31];
Para lograr generar las gráficas adecuadas. para poder observar la simulación, primero generamos mediante la función SurfaceGrafics una gráfica que contenga el	(* Se define el espacio tridimensional en que se graficará el manipulador, que se usará como gráfica base *)
espacio donde se mostrará el manipulador con una superficie simulando el piso y con la indicación de los ejes coordenados, la cual	(* Suma de la longitud del eslabón 2 y 3 *) L23 = L2+L3;
se almacena inicialmente en las variables g1 y g2.	g1=g2=SurfaceGraphics[{{0,0,0},{0,0,0},{0,0,0}}, MeshRange->{{-(L23),L23},{-(L23),L23}}, AxesLabel->{Eje X,Eje Y,Eje Z},
Definimos los parámetros de rotación de cada eslabón, en función de los ángulos que utilizaremos para girar cada uno de los eslabones que serán:	DefaultFont->{"Negrita", 12}, PlotRange->{{-(L23+.1),(L23+.1)}, {-(L23+.1),(L23+.1)}, {(1,1,(L23+.1)), /(1,1,(L23+.1))}
Eslabón 1 -> theta1	BoxRatios->{1,1,1},Axes->True];
Eslabón 2 \rightarrow theta2 Eslabón 3 \rightarrow theta3	(* Se inicializan en cero los ángulos de rotación de los tres eslabones *)
Se generan dos ciclos <i>For</i> , el primero (variable de control i) servirá para controlar cual eslabón deberá moverse, o sea, cuándo variar que ángulo, lo cual se logra utilizando la función <i>Switch</i> , y el segundo ciclo (variable de control j) nos permitirá generar varias gráficas del movimiento de cada eslabón. La variable i toma valores de 1 a 3, ya que son tres eslabones y la variable j toma valores de 1 a n, donde n es el número de gráficas a generar por cada eslabón.	<pre>theta1 = 0; theta2 = 0 ;theta3 = 0; (* Se definen los valores de los parámetros de las rotaciones en función de los ángulos de rotación*)</pre>
Una vez que los parámetros de rotación tienen un valor numérico, se evalúan numéricamente las ecuaciones del manipulador, haciendo uso del respaldo de ecuaciones realizado anteriormente	(* Se inicializa un ciclo para mover los tres eslabones, uno a la vez *) For[i=1, i<=3, i++,
Con las ecuaciones, ya numéricas, se modifica la gráfica almacenada en g2	(* Se inicializa otro ciclo para generar varias gráficas para cada eslabón en diferentes posiciones*)
agregándole un punto en la posición que corresponde a rp.	rorij=1, j<=0, j**, /* Se seleccional modiante la función Switch, cuel
De la misma forma, con las ecuaciones numéricas, se determinan los diferentes puntos de la gráfica a generar, es decir, se genera un punto para cada inicio y final de	eslabón se moverá según el valor de la variable i, lo cual se logra cambiando sus parámetros de rotación *)
cada esiabón, además de generar puntos	Switch[i,3, (* Se mueve el primer eslabón *) theta1+=15;

para mostrar el vector e11 de la base del primer eslabón. Con los puntos anteriores, se genera la gráfica correspondiente a la posición del manipulador, utilizando para ello la función <i>Graphics3D</i> , que nos permite trazar líneas de diferente color y espesor entre los puntos determinados.	 r10=N[Cos[theta1/2 Degree]]; r13=N[Sin[theta1/2 Degree]]; 2. (* Se mueve el segundo eslabón *) theta2+=15; r20=N[Cos[theta2/2 Degree]]; r21=N[Sin[theta2/2 Degree]]; 1. (* Se mueve el tercer eslabón *) r30=N[Cos[theta3/2 Degree]]; r31=N[Sin[theta3/2 Degree]]; theta2+=5; l;
Finalmente se muestran en la pantalla, de manera combinada, las gráficas g1, g2 y g3 que se generan en cada pasada de los ciclos, utilizando para esto la función <i>Show</i> la cual envía a la pantalla objetos gráficos y permite que se combinen varias gráficas.	(* Se asignan valores a las ecuaciones del manipulador, según los valores de los parámetros, utilizando el respaldo de ecuaciones*)
Se muestran, a manera de ejemplo, algunas gráficas de un proceso.	rp=N[rx];
	(* Se agrega a la grafica base la posición del elemento terminal como un punto *)
	g2={Graphics3D[{PointSize[0.05], Point[rp],RGBColor[0,1,0]}],g2};
	(* Longitud del vector que servirá de base del manipulador *)
	a= 3;
	(* Se definen y calculan los puntos en el espacio tridimensional para graficar el manipulador *)
	punto1 = {0,0,0}; punto2 = N[a*base]; punto3 = N[-a*base]; punto4 = N[b1]; punto5 = N[punto4 + b2];
	punto6 = N[punto5 + b3];
	(* Se genera la gráfica del manipulador en la posición correspondiente *)
	g3=Graphics3D[{
	(* Base rotatoria *) {AbsoluteThickness[40],RGBColor[0,.5,0], Line[{punto2,punto3}]],
	(* Esiabon 1 *) {AbsoluteThickness[25],RGBColor[0,1,0], Line[{punto1,punto4}]],
	<pre>{AbsoluteThickness[20],RGBColor[1,0,0], Line[{punto4,punto5}]}.</pre>

SIMULACION COMPUTACIONAL DE LA CINEMATICA



SIMULACION COMPUTACIONAL DE LA CINEMATICA



4.1.2 Cinemática Inversa.

La cinemática inversa del robot de tres grados de libertad, como se mencionó anteriormente, consiste en que, dada la posición del elemento terminal, se deben determinar los ángulos que deberán desplazarse cada una de las juntas del manipulador para alcanzar la posición deseada. En este caso, consiste en que dada la configuración no deformada (P_0 , P_1 , Q_0 , Q_1 ,) y la posición del elemento terminal ($r_p = \{r_px, r_py, r_pz\}$), se deben determinar los parámetros r1₀, r1₃, r2₀, r2₁, r3₀, r3₁ que satisfagan la ecuación de la posición del elemento terminal, que es la siguiente:

$$r_{p} = \{-4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}L_{3}), -2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}L_{3}), L_{1} + B1^{*}L_{2} + C2^{*}L_{3}\}$$
(4.9)

Para resolver la cinemática inversa del robot de tres grados de libertad, debemos plantear un sistema de 6 ecuaciones que nos permitan determinar los 6 parámetros de rotación buscados. De la ecuación de r_p obtenemos 3 ecuaciones y con las normas unitarias de los Quaterniones se completa el sistema, el cual se muestra a continuación.



4.1.2 Cinemática Inversa.

La cinemática inversa del robot de tres grados de libertad, como se mencionó anteriormente, consiste en que, dada la posición del elemento terminal, se deben determinar los ángulos que deberán desplazarse cada una de las juntas del manipulador para alcanzar la posición deseada. En este caso, consiste en que dada la configuración no deformada (P_0 , P_1 , Q_0 , Q_1 ,) y la posición del elemento terminal ($r_p = \{r_px, r_py, r_pz\}$), se deben determinar los parámetros r1₀, r1₃, r2₀, r2₁, r3₀, r3₁ que satisfagan la ecuación de la posición del elemento terminal, que es la siguiente:

$$\mathbf{r}_{a} = \{-4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}L_{3}), -2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}L_{3}), L_{1} + B1^{*}L_{2} + C2^{*}L_{3}\}$$
(4.9)

Para resolver la cinemática inversa del robot de tres grados de libertad, debemos plantear un sistema de 6 ecuaciones que nos permitan determinar los 6 parámetros de rotación buscados. De la ecuación de r_p obtenemos 3 ecuaciones y con las normas unitarias de los Quaterniones se completa el sistema, el cual se muestra a continuación.

 $r_n x = -4^*A5^*(B2^*L_2 + C1^*L_3)$ r.y= - 2*A6*(B2*L, + C1*L₃) $r_{0}z = L_{1} + B1^{*}L_{2} + C2^{*}L_{3}$ (4.10) $r1_0^2 + r1_3^2 = 1$ $r2_0^2 + r2_1^2 = 1$ $r3_{0}^{2} + r3_{1}^{2} = 1$ Siendo: A1 = P.*P.; A2 = 1-2"P."; A3 = Q, Q;; A4 = 1-2"Q.2: A6 = 1-2"r1,"; A5 = r1, 'r1; A8 = 1-2"r2,"; A7 = r2, "r2,; A9 = r3,"r3; A10 = 1-2"r3,"; (4.11) B2 = A2'A7 + A1'A8; B1 = A2*A8 - 4*A1*A7; B3 = A10*A8 - 4*A7*A9; B4 = A10*A7 + A8*A9; B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4: C1 = B4*85 + B3*86: C2 = B3*B5 - 4*B4*B6;

Lo anterior nos permite ubicar el elemento terminal en el punto {r_px, r_py, r_pz} dentro de su espacio de trabajo, ahora si deseamos que el elemento terminal siga una trayectoria determinada, una de las cosas que podemos hacer es discretizar dicha trayectoria en varios puntos a lo largo de ella y determinar los parámetros de rotación de los eslabones del manipulador para cada punto, resolviendo el sistema de ecuaciones (4.11), lo que nos permitirá ubicar el elemento terminal sobre dicha secuencia de puntos. Lo anterior no permite ubicar el elemento terminal en cualquier punto de la trayectoria, solo en los puntos en que dicha trayectoria se discretizó. A continuación se presenta el programa desarrollado en Mathematica³ para la cinemática inversa del manipulador de tres grados de libertad, el cual es continuación del anterior donde se presenta la cinemática directa.

Este programa es parte integral del anterior, es decir, para que funcione deberá tenerse la parte anterior donde se presenta la modelación y la cinemática directa, ya que se requieren las ecuaciones del manipulador. En esta sección se presenta el cálculo para tres diferentes trayectorias y desde luego la graficación para observar la simulación.	CINEMATICA INVERSA Solución de ecuaciones para diferentes trayectorias
En esta parte se determinan los parámetros de rotación de los eslabones, para una sucesión de puntos que pertenecen a una línea recta, la cual se define por sus dos puntos extremos(S, S1). Para lo anterior, primeramente recuperamos las ecuaciones del manipulador.	LINEA RECTA (* Se inicializan variables *) ClearAll[r10, r13, r20, r21, r30, r31, "solp", "solq", "solr", "solucion"];
las cuales se respaldaron en otras variables, (deberá resolverse la cinemática inversa para determinar la configuración no deformada y ejecutar la celda donde están las ecuaciones	(*Se recuperan las ecuaciones del manipulador*) b1=z1; b2=z2; b3=z3; rp=rx; rp1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]];
optimizadas para la configuración no deformada).	(* Variables de control *) t = 0; (* Para discretizar en varios puntos la trayectoria seleccionada *)

de la trayectoria *)

Para determinar la puntos sobre la trayectoria. | i = 1; (* Para guardar la solución de cada punto utilizamos la forma paramétrica de una recta que pasa por dos puntos $p_1 = \{x_1, y_1, z_1\} y$ $p_2 = \{x_1, y_1, z_1\}$

$$px = x_1 + (x_2 - x_1)t$$

$$py = y_1 + (y_2 - y_1)t$$

$$pz = z_1 + (z_2 - z_1)t$$

Con lo anterior se realiza un ciclo utilizando como variable de control del mismo, a la variable t, con lo que determinamos puntos sobre la trayectoria a seguir y resolvemos el sistema de ecuaciones (4.1).

$$r_{p}x = -4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}L_{3}),$$

$$r_{p}y = -2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}L_{3}),$$

$$r_{p}z = L_{1} + B1^{*}L_{2} + C2^{*}L_{3},$$

$$r1_{0}^{2} + r1_{3}^{2} = 1,$$

$$r2_{0}^{2} + r2_{1}^{2} = 1,$$

$$r3_{0}^{2} + r3_{1}^{2} = 1.$$

Este sistema de ecuaciones se resueive utilizando la función Findroot usando los valores iniciales mostrados. Una vez que se resueive el sistema de ecuaciones anterior, se verifica si la precisión de la solución es la adecuada, si es así, se fijan como valores iniciales, para resolver la cinemática inversa del siguiente punto, a la solución obtenida y se continua el ciclo, si la solución está fuera de la precisión se envía un mensaje y se restauran los valores iniciales continuando el ciclo hasta terminar.

(* Valores iniciales para resolver el sistema de ecuaciones *) ir10 = .7071; ir13 = .7071; ir20 = .3; ir21 = .9; ir30 = .5; ir31 = .8; (* Trayectoria en línea recta dada por los puntos extremos *) $S0 = \{5,5,7\}; S1 = \{5,-5,7\};$ (* Se inicia un ciclo para resolver el problema cinemático inverso en los puntos en que será discretizada la travectoria *) Forit=0, t<1, t+=.1, (* Se determinan las coordenadas del punto correspondiente de la trayectoria*) px=S0[[1]]+(S1[[1]]-S0[[1]])*t; py=S0[[2]]+(S1[[2]]-S0[[2]])*t; pz=S0[[3]]+(S1[[3]]-S0[[3]])*t; (* Solución del problema cinemático inverso para el punto seleccionado *) solucion[i] = FindRootl{rp1 == px, rp2 == py, rp3 == pz, r10^2+r13^2 == 1, r20^2+r21^2 == 1, r30^2+r31^2 == 1}, {r10,ir10},{r13,ir13},{r20,ir20}, {r21,ir21},{r30,ir30},{r31,ir31}, MaxIterations->50]; (* Se verifica la precisión de la solución *) e = 10^-5; (* Precisión requerida *) [f[Abs[px-(rp1/.solucion[i])] < e && Abs[py-(rp2/.solucion[i])]< e && Abs[pz-(rp3/.solucion[i])] < e, (* Si la solución está en la precisión fijada, se definen como valores iniciales, para resolver la cinemática inversa del siguiente punto de la travectoria a los valores de la solución obtenida. si no es así se restauran los valores iniciales utilizados *)

ir10=r10/.solucion[i]; ir13=r13/.solucion[i]; ir20=r20/.solucion[i]; ir21=r21/.solucion[i]; ir30=r30/.solucion[i]; ir31=r31/.solucion[i],

	B. M. MARTINE, M.
	(* Se imprime un mensaje si la solución no está en la precisión definida *)
	Print["No se encontró solución para el punto:", {px,py,pz}];
	(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
	ir10=.7071; ir13=.7071; ir20=.3; ir21=.9; ir30=.5; ir31=.8];
	(* Se incremente la variable i y se cierra el ciclo For *)
	i++]
En esta sección, al igual que en la anterior se	CIRCUNFERENCIAS O ELIPSES
resuelve la cinemática inversa para una	(* Se inicializan variables *)
que en este caso se trata de una elipse o circunferencia. El proceso que se sigue es el	ClearAll[r10, r13, r20, r21, r30, r31, "solp", "solq", "solr", "solucion"];
mismo que para la línea recta, solamente cambia la manera de determinar los puntos	(*Se recuperan las ecuaciones del manipulador*)
sobre la trayectoria, que en este caso se	b1=z1; b2=z2; b3=z3; rp=rx; rp1=rp[[1]];
realiza utilizando la forma paramétrica de una	rp2=rp[[2]]; rp3=rp[[3]];
con radios radio1 y radio2.	(* Variables de control *)
px = h;	t = 0; (* Para discretizar en varios puntos la
py = radio1 * Cos[t] +k ; pz = radio2 * Sin[t] + l;	trayectoria seleccionada *) i = 1: (* Para quardar la solución de cada punto
	de la trayectoria *)
!	(*Variables de control para definir la trayectoria *)
{h, k, l}	radio1 = 7; (* Radio uno de la elipse *) radio2 = 5; (* Radio dos de la elipse *) h=5;k=0;l=6; (* Centro de la elipse *)
radio2 radio1	(* Valores iniciales para resolver el sistema de ecuaciones *)
	ir10 = .7071; ir13 = .7071; ir20 = .3; ir21 = .9; ir30 = .5; ir31 = .8;
·	(* Se inicia un ciclo para resolver el problema cinemático inverso en los puntos en que será discretizará la trayectoria *)
	For[t=0, t<=2*N[Pi], t+=N[Pi]/9,

```
(* Se determinan las coordenadas del punto
  correspondiente de la trayectoria *)
      px = h;
      py = radio1*Cos[t] + k;
      pz = radio2*Sin[t] + I;
(* Solución del problema cinemático inverso para
  el punto seleccionado *)
solucion[i] =
FindRoot[{rp1 == px, rp2 == py, rp3 == pz,
          r10^2+r13^2 == 1, r20^2+r21^2 == 1,
          r30^2+r31^2 == 1).
          {r10,ir10},{r13,ir13},{r20,ir20},
          {r21,ir21},{r30,ir30},{r31,ir31},
          MaxIterations->50];
(* Se verifica la precisión de la solución *)
e = 10^-5; (* precisión requerida *)
If[Abs[px-(rp1/.solucion[i])] < e &&
 Abs[py-(rp2/.solucion[i])] < e &&
 Abs[pz-(rp3/.solucion[i])] < e,
(* Si la solución está en la precisión fijada se
  definen como valores iniciales, para resolver la
  cinemática inversa del siguiente punto de la
  travectoria, a los valores de la solución obtenida.
  si no es así se restauran los valores iniciales
  utilizados *)
ir10=r10/.solucion[i]; ir13=r13/.solucion[i];
ir20=r20/.solucion[i]; ir21=r21/.solucion[i];
ir30=r30/.solucion[i]; ir31=r31/.solucion[i],
(* Se imprime un mensaje si la solución no está en
  la precisión definida *)
Print("No se encontro solución para el punto:",
    {px,py,pz}];
(* Se restauran valores iniciales para resolver las
  ecuaciones si no se encontró solución para el
  punto en cuestión *)
ir10=.7071; ir13=.7071; ir20=.3; ir21=.9;
ir30=.5; ir31=.8];
(* Se incremente la variable i y se cierra el ciclo
  For *)
        1++1
```


	If[Abs[px-(rp1/.solucion[i])] < e &&
	Abs[py-(rp2/.solucion[i])] < e &&
	Abs[pz-(rp3/.solucion[i])] < e.
	/* Si la solución está en la precisión filada se
	dennen como valores iniciales, para resolver la
	cinematica inversa del siguiente punto de la
	trayectoria, a los valores de la solución obtenida,
	si no es así se restauran los valores iniciales
	utilizados *)
	CONTRACTOR AND AND ADDRESS AND ADDRESS
	ir10=r10/.solucion[i]; ir13=r13/.solucion[i];
	ir20=r20/.solucion[i]; ir21=r21/.solucion[i];
	ir30=r30/.solucion[i]: ir31=r31/.solucion[i].
	/* Se imprime un mensaie si la solución no está en
	la presizión definida *)
	a precision dennida)
	Printi No se encontro solución para el punto: ",
	{px,py,pz}];
	(* Se restauran valores iniciales para resolver las
	ecuaciones si no se encontró solución para el
	punto en cuestión *)
	ir10=.7071; ir13=.7071; ir20=.3; ir21=.9;
	ir30= 5: ir31= 81:
	/* Se incremente la variable i v se cierra el ciclo
	Eor *)
	ا ج ج ا
Lina vez determinados los parámetros de	Graficación para visualizar la simulación de la
rotación para la travestoria seleccionada	travectorio seleccionede
abore se constante las artificas que normitirán	trayectoria seleccionada.
anora se generaran las grancas que permituran	Class Allfat0 at2 a20 a24 a20 a241
ver la simulación de la cinematica inversa del	GlearAll[r10,r13,r20,r21,r30,r31]
manipulador de tres grados de libertad. Para	
poder realizarlo, recuperamos los valores de	("Se recuperan los parametros de rotación
los parámetros almacenados en la variable	almacenados en la variable solucion[j], obtenidos
solución[j] y los almacenamos en variables	de resolver el problema cinemático inverso y se
separadas para cada par de parámetros. Al	almacenan en variables para cada par de
igual que en la cinemática inversa, se genera	parámetros *)
una gráfica que servirá como base para	
graficar el manipulador, la cual se almacena en	For[j=1, j <i, j++,="" solr1[j]="{r10,r13}/.solucion[j];</th"></i,>
las dos variables g1 y g2, posteriormente se	solr2[j]={r20,r21}/.solucion[j];
inicia un ciclo, que se repetirá tantas veces	solr3[j]={r30.r31}/.solucion[j]]
como valores de los parámetros se havan	
calculado para la travectoria seleccionada	(* Se define el espacio tridimensional en que se
(variable "i") En cada nasada dal ciclo sa	oraficará el manipulador que la llamaremos
asianan los valores correspondientes a los	Arenderer of memberedor, deo le membrenioa
	aráfica base *)
asignan los valores conespondientes a los	gráfica base *)
parámetros y con ellos se genera la gráfica del	gráfica base ")
parámetros y con ellos se genera la gráfica del manipulador de la misma forma que se hizo en	gráfica base *) (* Suma de la longitud del eslabón 2 y 3 *)

de ejemplo, trayectoria.	algunas	gráficas	de	una	g1=g2=SurfaceGraphics[{{0.0.0},{0.0.0},{0.0.0}}, MeshRange->{{-(L23),L23},{-(L23),L23}}, AxesLabel->{Eje X,Eje Y,Eje Z}, DefaultFont->{"Negrita",12}, PlotRange->{{-(L23+.1),(L23+.1)},
					(* Se inicia un ciclo para generar las gráficas de cada punto de la trayectoria *)
					For[j=1, j <i, j++,<="" td=""></i,>
					(* Se asigna el valor de los parámetros para el punto de la trayectoria correspondiente almacenados en las variables solr?[j]*)
					r10=solr1[j][[1]]; r13=solr1[j][[2]]; r20=solr2[j][[1]]; r21=solr2[j][[2]]; r30=solr3[j][[1]]; r31=solr3[j][[2]];
					(* Se asignan valores a las ecuaciones del manipulador, según los valores de los parámetros, utilizando el respaldo de ecuaciones *)
					base=N[a11];
					(* Longitud del vector que servirá de base del manipulador *)
					a = 3;
					(* Se definen los puntos en el espacio tridimensional para graficar el manipulador *)
					punto1 = {0,0,0}; punto2 = N[a*base]; punto3 = N[-a*base]; punto4 = N[b1]; punto5 = N[punto4 + b2]; punto6 = N[punto5 + b3];
					(* Se agrega a la gráfica base (g2) la posición del elemento terminal, como un punto *)
					g2={Graphics3D[{{PointSize[0.01], Point[rp]}}], g2};
					(* Se genera la gráfica del manipulador en la posición correspondiente *)



SIMULACION COMPUTACIONAL DE LA CINEMATICA



4.2 Cinemática de un robot de cutro grados de libertad.

En la siguiente figura se muestra la configuración del robot que se simulará en está sección, el cual fue previamente modelado en el capítulo amirior.



Figura 4.2 Configuración demanipulador de cuatro grados de libertad

4.2.1 Cinemática Directa.

El problema cinemático directo para este unipulador, consiste en que dados los desplazamientos angulares θ_1 , θ_2 , θ_3 y θ_4 , se debe determinar la posición y orientación del elemento terminal, asl como la configuración resultante de sus dabones. Para realizar lo anterior hacemos uso de las ecuaciones (3.85), (3.86), (3.87) determinantes en el capitulo anterior, que al evaluarlas se resuelve la cinemática directa de este manipulador. Enclas ecuaciones se muestran a continuación.

$$b_{1} = \{0, 0, L_{1}\}$$

$$b_{2} = \{-4^{*}A5^{*}B2^{*}L_{2}, -2^{*}A6^{*}B2^{*}L_{2}, 32^{*}L_{2}\}$$

$$b_{3} = \{-4^{*}A5^{*}C1^{*}L_{3}, -2^{*}A6^{*}C1^{*}L_{2}, 32^{*}L_{3}\}$$

$$b_{4} = \{-4^{*}A5^{*}C1^{*}L_{4}, -2^{*}A6^{*}C1^{*}L_{2}, 32^{*}L_{3}\}$$

$$b_{5} = \{(A12^{*}A6 - 4^{*}A11^{*}A5^{*}C2)^{*}L_{2}, -2^{*}(A12^{*}A5 + A11^{*}A6^{*}C2)^{*}L_{4}, -4^{*}A11^{*}C1^{*}L_{4}\};$$

$$r_{p} = \{-4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}(L_{3} + L_{4})),$$

$$-2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}(L_{3} + L_{4})), 4_{1} + B1^{*}L_{2} + C2^{*}(L_{3} + L_{4})\}$$
Donde:
$$A1 = P_{0}^{*}P_{1}; \quad A2 = 1-2^{*}P_{0}^{2};$$

$$A3 = Q_{0}^{*}Q_{1}; \quad A4 = 1-2^{*}Q_{0}^{2};$$

 $A5 = r1_0^{\circ} r1_3; \quad A6 = 1-2^{\circ} r1_0^{-2};$ $A7 = r2_0^{\circ} r2_1; \quad A8 = 1-2^{\circ} r2_0^{-2};$ (4.14) Las variables de las ecuaciones anteriores tienen el siguiente significado:

 $\begin{array}{l} \mathsf{L}_1, \mathsf{L}_2, \mathsf{L}_3, \mathsf{L}_4, \mathsf{L}_5 \rightarrow \mathsf{Longitud} \ \text{de los eslabones del manipulador.} \\ \mathsf{b}_1, \mathsf{b}_2, \mathsf{b}_3, \mathsf{b}_4, \mathsf{b}_5 \rightarrow \mathsf{Eslabones del manipulador.} \\ \mathsf{r}_p \rightarrow \mathsf{Posición del elemento terminal.} \\ \mathsf{P}_0, \mathsf{P}_1, \mathsf{Q}_0, \mathsf{Q}_1 \rightarrow \mathsf{Parámetros de rotación en la configuración no deformada.} \\ \mathsf{r1}_0, \mathsf{r1}_3, \mathsf{r2}_0, \mathsf{r2}_1, \mathsf{r3}_0, \mathsf{r3}_1, \mathsf{r4}_0, \mathsf{r4}_3 \rightarrow \mathsf{Parámetros de rotación en la configuración no deformada.} \end{array}$

Los parámetros de rotación tienen la siguiente relación con las rotaciones de los eslabones del manipulador.

$P_0 = Cos (\phi_1/2);$ $Q_0 = Cos (\phi_2/2);$	$P_1 = Sen (\phi_1/2);$ $Q_1 = Sen (\phi_2/2);$	(4.15)
r1, = Cos (0,/2);	r1, = Sen (θ,/2);	
12, = Cos (0,/2);	$r_{2_{1}} = Sen(\theta_{2}/2);$	(4.16)
r3, = Cos (0,/2);	$r_{3} = Sen(\theta_{3}/2);$	
$r4_{0} = \cos{(\theta_{4}/2)};$	$r4_3 = Sen (\theta_4/2);$	

Siendo:

φ₁ el ángulo de rotación del eslabón 2 en la configuración no deformada.φ₂ el ángulo de rotación del eslabón 3 en la configuración no deformada.

θ, el ángulo de rotación del eslabón 1 en la configuración deformada.

θ₂ el ángulo de rotación del eslabón 2 en la configuración deformada.

 θ_3 el ángulo de rotación del eslabón 3 en la configuración deformada.

θ₄ el ángulo de rotación de la pinza en la configuración deformada.

Antes de poder realizar el proceso de cinemática directa del manipulador, debemos determinar la configuración no deformada del robot mediante un proceso de cinemática inversa, que consiste en calcular los parámetros $P=\{P_0, P_1, 0, 0\}$ y $Q=\{Q_0, Q_1, 0, 0\}$ dada la posición del elemento terminal R_p , que en este caso es el extremo del eslabón 3 donde esta unida la pinza (Ecuaciones (3.71), (3.72)). Es decir, con las siguientes ecuaciones puede plantearse un sistema de ecuaciones, como se muestra.

Donde:

A1 =
$$P_0^*P_1$$
; A2 = 1 - 2* P_0^2 ; (4.18)
A3 = $Q_0^*Q_1$; A4 = 1 - 2* Q_0^2 ;

Con la ecuación para R_p podemos plantear 2 ecuaciones si damos la posición para el elemento terminal como R_p = {0, R_py, R_pz} y con las normas unitarias de los Quaterniones P y Q completamos un sistema de 4 ecuaciones con 4 incógnitas, que al resolverlo obtenemos los parámetros que determinan la configuración no deformada.

2'((A2'A3 + A1'A4) *L₃ - A1'L₂) =
$$R_p y$$

L₁ - A2*L₂ + (A2*A4 - 4*A1*A3)*L₃ = $R_p z$ (4.19)
 $P_0^2 + P_1^2 = 1$
 $Q_0^2 + Q_1^2 = 1$

Con lo anterior podemos ahora resolver el problema cinemático directo y realizar la correspondiente simulación computacional, que como se mencionó se realizará en Mathematica[®]. A continuación se presenta el programa desarrollado para este caso. Con el objeto de mostrar la forma de utilizar Mathematica[®] para obtener las ecuaciones del manipulador, tanto en su configuración no deformada como en la deformada.

En la primera parte del programa se define la transformación rotación de un vector	Definición de la Transformación Rotación.
tridimensional y de un Quaternión, y de algunas otras funciones necesarias que se	(* Se apagan algunos mensajes del sistema *)
muestran a continuación.	Off[General::speil1] Off[General::speil]
Transformación de un Quaternión en un	
vector tridimensional.	(* Se define la Transformación de un Quaternión a un vector tridimensional Tv[] *)
$q = \{q_0, q_1, q_2, q_3\} \in Q$	and the residues many frances
	Tv[q_]:={q[[2]],q[[3]],q[[4]]}:
$Tv[\mathbf{q}] = \{\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3\}.$	(* Se define la Transformación inversa de Tv[] *)
Transformación vectorial inversa.	TvInv[v_]:={0,v[[1]],v[[2]],v[[3]]};
$v = \{v_1, v_2, v_3\} \in \Re^3$	(* Se define la Transformación de un Quaternión a Quaternión vectorial *)
$Tvinv[v] = \{0, v_1, v_2, v_3\}.$	TvQ[q_]:={0,q[[2]],q[[3]],q[[4]]};
Transformación de un Quaternión en un	
Quaternión vectorial.	(* Se define la Multiplicación entre dos Quaterniones *)
$TvQ[q] = \{0,q_1, q_2, q_3\}.$	
	ProductoQ[p_,q_]:=
Producto de dos Quaterniones.	$ \begin{array}{l} \left\{ p[(1]]^{\circ}q[(1]] \ - \ p[(2]]^{\circ}q[(2]] \ - \ p[(3]]^{\circ}q[(3]] \ - \ p[(4]]^{\circ}q[(4]], \\ p[(1]]^{\circ}q[(2]] \ + \ p[(2]]^{\circ}q[(1]] \ + \ p[(3)]^{\circ}q[(4]] \ - \ p[(4]]^{\circ}q[(3]). \end{array} \right. $
$p = \{p_0, p_1, p_2, p_3\}, q = \{q_0, q_1, q_2, q_3\} \in Q$	$p[(1)]^q[(3)] - p[(2)]^q[(4)] + p[(3)]^q[(1)] + p[(4)]^q[(2)],$ $p[(1)]^q[(4)] + p[(2)]^q[(3)] - p[(3)]^q[(2)] + p[(4)]^q[(1)]);$
ProductoQ[p,q] =	
{p ₀ *q ₀ - p ₁ *q ₁ - p ₂ *q ₂ - p ₃ *q ₃ ,	(* Se define el conjugado de un Quaternión *)
$p_0^*q_1 + p_1^*q_0 + p_2^*q_3 - p_3^*q_2$	have a stranger of the state of
po*q2 - p1*q3 + p2*q0 + p3*q1,	ConjugadoQ[p_]:={p[[1]],-p[[2]],-p[[3]],-p[[4]]};
Po*q3 + P1*q2 - P2*q1 + P3*q0)	
	(* Se define la Transformación Rotación de un
Conjugado de un Quaternión.	vector tridimensional *)
ConjugadoQ[p] = { p ₀ , - p ₁ , - p ₂ , - p ₃ }	Rotacion[p_,v_]:=
Rotación de un vector v ∈ ℜ³ con un	ConjugadoQ[p]]]];

Quaternión p, siendo p = 1. Según la	(* Se define la Transformación Rotación de un
definición (2.13).	Quaternion *)
$\rho(\mathbf{p},\mathbf{q}) = \frac{1}{\ \mathbf{p}\ ^2} \bullet \left(\mathbf{p}^* \mathbf{q}^* \overline{\mathbf{p}}\right), \forall \mathbf{q} \in \mathbf{Q}$	TvQ[ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]];
Rotacion[p,v] = Tv[p * Tvlnv[v] * p]	
Rotación de $\mathbf{q} \in \mathbf{Q}$ con un Quaternión p. siendo $\ \mathbf{p}\ = 1$. De acuerdo con la definición	
(2.13) y considerando que el primer elemento de q no cambia al rotar cuando $\ \mathbf{p}\ = 1$.	
RotacionQ[p,q] = {q₀,0,0,0} + TvQ[p*q* p̄]	
Se realizan definiciones iniciales necesarias.	Definiciones iniciales
como son la base canónica de 93° y las bases que servirán como bases locales en los	(* Base canónica *)
eslabones. También se define la forma local de los Quaterniones que nos permitirán rotar,	e1 = {1,0,0}; e2 = {0,1,0}; e3 = {0,0,1};
deformada.	(* Bases locales de los esiabones *)
	(* Eslabón 1 *) e1 = {1,0,0}; e2 = {0,1,0}; e3 = {0,0,1};
	(* Esiabón 2 *) f1 = {1,0,0}; f2 = {0,1,0}; f3 = {0,0,1};
	(* Eslabón 3 *) g1 = {1,0,0}; g2 = {0,1,0}; g3 = {0,0,1};
	(* Eslabón 4 *) h1 = {1,0,0}; h2 = {0,1,0}; h3 = {0,0,1};
	(*Quaterniones para rotar a posición no deformada*)
	$P = \{P0, P1, 0, 0\};$ (* Estabón 2 *) $Q = \{Q0, Q1, 0, 0\};$ (* Estabón 3 *)
	(* Quaterniones para rotar a posición deformada *)
	r1 = {r10, 0, 0, r13}; (* Eslabón 1 *)
	r2 = {r20, r21, 0, 0}; (* Eslabón 2 *)
	r3 = {r30, r31, 0, 0}; (* Eslabón 3 *)
	r4 = {r40, 0, 0, r43}; (* Eslabón 4 *)
En esta seccion del programa se realiza la	MODELADO DE LA CONFIGURACION NO
Modelación de la consiguración no deformada	UEFORMADA
dei manipulador. Como la forma de las bases	Modelo pero la configuración no deformado
Invales de los estabolies, en la contiguración	modelo para la configuración no deformada

no deformada, se obtuvo en el capitulo	(* Se inicializan variables *)
del las ecuaciones del manipulador	
simplemente obtenemos de manera directa la	(* Se definen variables para las normas unitarias de
forma de las bases locales, que son	los Quaterniones P v Q *)
	normas1= $(P1->(1-P0^{2})^{(1/2)}, Q1->(1-Q0^{2})^{(1/2)};$
$f^{\dagger} = a(\mathbf{P}, \mathbf{f})$ (Eclober 2)	normas2= ${(1-P0^{2})^{(1/2)} > P1, (1-Q0^{2})^{(1/2)} > Q1}$
$\mathbf{r}_j = p(\mathbf{r}_j \mathbf{r}_j)$ (Estaboli 2)	
$\mathbf{g}_{i}^{II} = \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{g}_{i})$ (Eslabon 3)	(* Se obtienen los productos de Quaterniones
h = h	necesarios para rotar las bases *)
$\mathbf{n}_j = p((\mathbf{P} \cdot \mathbf{Q}), \mathbf{n}_j)$ (Pinza)	
encourrent and the second	PQ = ProductoQ[P,Q];
Para lo anterior es necesario definir las	
normas unitarias de los Quaterniones P y Q.	(* Se obtiene la forma de las bases locales de los
las cuales se almacenan en las variables	eslabones en la configuración no deformada ")
normas i y normasz, para aplicarias en las	
ecuaciones resultantes de las rotaciones.	
l as variables que se utilizan nara las hases	$ e_1 - \{1, 0, 0\}, e_2 - \{0, 1, 0\}, e_3 - \{0, 0, 1\},$
locales de los eslabones son:	(* Felabón 2 *)
	f11 = ExpandiExpandiRotacioniP f11// pormas11
ei = Primer eslabón.	//.normas2:
fi = Segundo eslabón.	f21 = Expand/Expand/Rotacion/P.f21//.normas11
gi = Tercer eslabón.	//.normas2;
hj = Pinza. j = 1,2,3.	f31 = Expand[Expand[Rotacion[P,f3]]//.normas1]
	//.normas2;
	(* Eslabás 3 *)
	(Esiabori 5) a12 = ExpandiExpandiRotacioniPO a111// pormae11
	// normas2
	o22 = Expand/Expand/Rotacion/PQ.o21//.normas1
	//.normas2;
	g32 = Expand[Expand[Rotacion[PQ,g3]]//.normas1]
	//.normas2;
	(* Esiabón 4 *)
	h12 = Expand[Expand]Rotacion[PQ.h1]]//.normas1]
	//.normas2;
	h22 = Expand[Expand[Rotacion[PQ,h2]]//.normas1]
	//.normas2;
	h32 = Expand[Expand[Rotacion[PQ,h3]]//.normas1]
	//.normas2;
Los asishanas dal maninuladas y la nasisián	(* Vectores que definen los estabones del
del elemento terminal con identificados en la	
nosición no deformada nor las variables	manipulator /
siouientes.	b1 = L1*e3; (* Eslabón 1 *)
	$b2 = L2^{+}(31)$; (* Eslabón 2 *)
b 1 = Esla bón 1.	$b_3 = L_3^{\circ}g_3^{\circ}2;$ (* Eslabón 3 *)
b2 = Eslabón 2.	b4 = L4*h32; (* Eslabón 4 (pinza)*)
b3 = Eslabón 3.	b5 = L5*h12; (* Eslabón 4 (pinza)*)
b4 = Eslabón 4.	
b5 = Eslabón 4.	(* Constantes que se repiten y que servirán para
	simplificar las ecuaciones y optimizar operaciones *)

Rp = b1+b2+b3; Rp = Posición del elemento terminal.	constantes = {P0*P1->A1, P0^2->(1-A2)/2, Q0*Q1->A3, Q0^2->(1-A4)/2};
Las ecuaciones se simplifican utilizando las siguientes constantes	b2 = b2//.constantes; b3 = Expand[Expand[b3//.constantes]//.constantes];
A1 = P0*P1; A2 = 1 - 2*P0^2;	D4 = Expand[Expand[D4//.constantes]//.constantes];
A3 = Q0"Q1; A4 = 1 - 2"Q0^2;	(*Vector que define posición del elemento terminal *)
	Rp = b1+b2+b3;
Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales	Ecuaciones optimizadas para la posición no deformada.
P y Q, que definen la configuración no deformada en una posición dada del	ClearAll[P0,P1,Q0,Q1]
elemento terminal, que será de donde arrangue la simulación de la cinemática	(* Ecuaciones optimizadas *)
directa o inversa del manipulador en	b1 = {0, 0, L1}; (* Eslabón 1 *)
cuestión. Es decir, antes de generar una	b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *)
simulación computacional de la cinemática directa o inversa del manipulador, debemos	b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Esiabón 3 *) (A2*A4 - 4*A1*A3)*L3};
determinar la configuración no deformada realizando un proceso de cinemática inversa	(* Pinza *)
conocida del elemento terminal la cual debe	$bA = 10 2^{\circ}(A2^{\circ}A3 + A1^{\circ}A4)^{\circ} A (A2^{\circ}A4 - A^{\circ}A1^{\circ}A3)^{\circ} A)^{\circ}$
tener la forma siguiente:	$b5 = \{L5, 0, 0\};$
Rp = {0,Rpy, Rpz}.	(* Posición el elemento terminal *)
	Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3};
	(* Constantes de las ecuaciones *)
	A1 = P0*P1; A2 = 1 - 2*P0^2;
	A3 = Q0°Q1; A4 = 1 - 2°Q0^2;
	(* Longitud de los eslabones del manipulador *)
	L1 = 6; L2 = 5; L3 = 4;
Se resuelve el sistema de ecuaciones	Cinemática Inversa para determinar la
mostrado en (4.19), para P0,P1,Q0,Q1,	Configuración no Deformada
utilizando la función FindRoot, cuya sintaxis	
se puede consulta en el Apendice, al igual	determinan los Quaterniones "P" v "O"
este trabaio.	correspondientes *)
Los resultados obtenidos se redondean si	
alguno de ellos está muy cercano a cero,	ClearAll[P0,P1,Q0,Q1]
finalmente se almacenan en las variables	
correspondientes (P0, P1, Q0, Q1) para	(* Se da la posición del elemento terminal *)
poder ser utilizados posteriormente, ademas,	

se imprimen dichos valores.	posicion = {0, 0, 15};
	(* Se determinan "P" y "Q" utilizando los siguiente valores iniciales *)
	iP0 = .7071; iP1 = .7071; iQ0 = .866; iQ1 = .5;
	(* Se almacena la solución en la variable sol y se utiliza la función FindRoot para determinar la solución del sistema de ecuaciones *)
	FindRoot[{Rp2 == posicion[[2]],Rp3 == posicion[[3]], P0^2+P1^2 == 1, Q0^2+Q1^2 == 1}, {P0,iP0}, {P1,iP1}, {Q0,iQ0}, {Q1,iQ1}, MaxIterations->50];
	(* Se asignan los valores de la solución a las variables correspondientes *)
	P0=P0/.soi; P1=P1/.soi; Q0=Q0/.soi; Q1=Q1/.soi;
	(* Se redondean los valores de la solución *)
	lf[Abs[P0]==1,P1=0,lf[Abs[P1]==1,P0=0]]; lf[Chop[Abs[Q0]-1]==0,Q0=1;Q1=0, lf[Chop[Abs[Q1]-1]==0,Q0=0;Q1=1]];
	(* Se imprimen los resultados *)
	Print["P0 = ",P0 ," ", "P1 = ", P1, " Q0 = ", Q0, " ", "Q1 = ", Q1]
De la misma forma que se realizó la	MODELADO DE LA CONFIGURACION
modelación de la configuración no	DEFORMADA
forma final de las bases locales de los	Modelo para la configuración deformada
capitulo anterior y son:	CiearAll[r10,r13,r20,r21,r30,r31,L1,L2,L3];
$\mathbf{e}_{j}^{I} = \rho(\mathbf{r1}, \mathbf{e}_{j})$	(* Posición no deformada conocida *)
$\mathbf{f}_{i}^{111} = \rho(\mathbf{r1}^*\mathbf{r2}^*\mathbf{P}, \mathbf{f}_{i})$	$P = \{P0, P1, 0, 0\}, Q = \{Q0, Q1, 0, 0\},$
$g_{j}^{V} = \rho(r1^{*}r2^{*}r3^{*}P^{*}Q, g_{j})$	(* Quaterniones para rotar a la configuración deformada *)
$h_{j}^{VI} = \rho(r1^{*}r2^{*}r3^{*}r4^{*}P^{*}Q, h_{j})$	r1 = {r10,0,0,r13}; (* Rotación Eslabón 1*)
_	$r_{3} = \{r_{3}, r_{3}, r_{3}, 0, 0\}; (* Rotación Esiabón 2*)$
También en este caso, se definen las normas	r4 = {r40,0,0,r43}; (* Rotación Eslabón 4*)
almacenándolas en las variables "normas3"	(* blormes unitering de les Queternienes *)
y "normas4", que incluyen las normas	(Normas Unitarias de los Quaterniones *)
unitarias de los Quaterniones	r13->(1-r10^2)^(1/2) r21->(1-r20^2)^(1/2)
correspondientes a la configuración no	r31->(1-r30^2)^(1/2), r43->(1-r40^2)^(1/2)}];

deformada.	normas4 = Flatten[{normas2,
	(1-r10^2)^(1/2)->r13, (1-r20^2)^(1/2)->r21,
	(1-r30^2)^(1/2)->r31, (1-r40^2)^(1/2)->r43}];
	(* Sa obtionen les productos de Quaterniones
	necesarios para rotar las bases *)
	r1r2P = ProductoQ[r1,ProductoQ[r2,P]];
	r1r2r3PQ = ProductoQ[r1,ProductoQ[r2, ProductoQ[r3,ProductoQ[P,Q]]]];
	r1r2r3PQr4 = ProductoQ[r1,ProductoQ[r2, ProductoQ[r3,ProductoQ[P, ProductoQ[Q,r4]]]]];
	(* Se obtiene la forma final de las bases locales de los eslabones *)
	(* Base local del eslabón 1 *)
	e11 = Expand[Expand[Rotacion[r1,e1]]//.normas3]
	e21 = Expand[Expand[Rotacion[r1,e2]]//.normas3]
	e31 = Expand[Expand[Rotacion[r1,e3]]//.normas3] //.normas4;
	(* Base local del eslabón 2 *)
	f13 = Expand[Expand[Rotacion[r1r2P,f1]]//.normas3]
	f23 = Expand[Expand[Rotacion[r1r2P,f2]]//.normas3]
	f33 = Expand[Expand[Rotacion[r1r2P,f3]]//.normas3] //.normas4;
	(* Base local del eslabón 3 *)
	g15 = Expand[Expand[Rotacion[r1r2r3PQ,g1]]
	g25 = Expand[Expand[Rotacion[r1r2r3PQ.g2]] // normas31// normas4;
	g35 = Expand[Expand[Rotacion[r1r2r3PQ,g3]] //.normas3]//.normas4;
	(* Base local del esiabón 4 *)
	h16 = Expand[Expand[Rotacion[r1r2r3PQr4,h1]]
	h26 = Expand[Expand[Rotacion[r1r2r3PQr4,h2]]
	h36 = Expand[Expand[Rotacion[r1r2r3PQr4,h3]] //.normas3]//.normas4;

Una vez obtenidas las bases locales en su	(* Vectores que definen los eslabones del
forma final, se obtienen las ecuaciones que	manipulador ")
representan los eslabones del manipulador y	D1 = L1*e31; (* Eslabon 1 *)
la posición del elemento terminal como.	$D2 = L2^{+}33;$ (* Esiabon 2 *)
	b3 = L3 g35, (Estabon 3') $b4 = L4^{*}b36^{*}$ (* Estabon 4 (pipza)*)
$D_1 = L_1 = C_3 = L_1 = C_1$; (Eslabor 1)	b5 = 1 5*h16; (* Eslabón 4 (pinza)*)
$b_2 = L_2^* f_3^{111} = L_2^* f_33;$ (Eslabón 2)	
$b_3 = L_3^* g_3^V = L_3^* g_3^S;$ (Eslabón 3)	(* Constantes que se repiten y que servirán para
$b_4 = L_4^* h_3^{VI} = L_4^* h_3^{0}$; (Pinza)	simplificar las ecuaciones y optimizar operaciones (
$b_s = L_s^* h_1^{VT} = L_s^* h_1 6;$ (Pinza)	constantes = $\{P0^{\circ}P1 - > A1, P0^{\circ}2 - > (1 - A2)/2, \\ Q1 - Q2 - Q2 - Q2 - 2 - (1 - A2)/2, \\ Q2 - Q2 - Q2 - Q2 - (1 - A2)/2, \\ Q2 - Q2$
(Destables det alemante terminal)	QU-Q1-2A3, QU-2-2(1-A4)/2,
(Posicion del elemento terminal)	F10*F13-2A0, F10*2-2(1-A0)/2,
	120121-287, 12012-2(1-80)/2,
$r_p = D_1 + D_2 + D_3 + D_4$	ran*ra3-> A11 ran^2->(1-A12)/2
Estas everacionas con simplificadas	A2*A9->4*A1*A7+B1 A2*A7->B2-A1*AB
eustituvendo por constantes a aquellos	A10*A9->4*A7*A9+B3 A10*A7->B4-A8*A9
términos que se reniten dentro de las	A2*A4->4*A1*A3+B5 A2*A3->B6-A1*A4
diferentes ecuaciones como se puede	A10*A4->4*A3*A9-B7 A4*A9->B8-A10*A3
observar.	B4*B5->C1 - B3*B6, B3*B5->C2+4*B4*B6);
	b2 - Expandib2// constantes);
	b2 - Expand(Collectib2 // 2 A5 A6))// constantes);
	b3 = Expand[b3//.constantes];
	b3 = Collect[Collect[Collect[b3,{L3,A5,A1,A2,A3,A4,
	A6}]//.constantes, {L3,A5,A6,B3,B4}]
	//.constantes,{L3,A5,A6,B3,B4,B5,B6}];
	b3 = Expand[b3//.constantes];
	b4 = Expand[b4//.constantes];
	b4 = Collect[Collect[Collect[b4,{L4,A5,A1,A2,A3,A4,
	A6}]//.constantes, {L4,A5,A6,B3,B4}]
	//.constantes,{L4,A5,A6,B3,B4,B5,B6}];
	b4 = Expand[b4//.constantes];
	b5 = Expand[b5//.constantes]:
	b5 = Collect[Collect[Collect[b5,{L4,A5,A1,A2,A3,A4.
	A6}]//.constantes, {L4,A5,A6,B3,B4}]
	//.constantes, {L4,A5,A6,B3, B4,B5,B6}];
	b5 = Expand[b5//.constantes];
	(*Vector que define posición del elemento terminal *)
	rp = Factor[b1 + b2 + b3 + b4];
En esta parte se presenta el resultado	Ecuaciones optimizadas para la posición
obtenido en la sección anterior, es decir, las	deformada (Secuencia 3-2-1)
ecuaciones simplificadas del manipulador,	
así como, las constantes utilizadas en dichas	CiearAil[r10, r13, r20, r21, r30, r31, r40, r43, L1, L2,
ecuaciones. Esta sección es la que	L3, L4, L5];
propiamente se utiliza para realizar la	

1		
	simulación de la cinemática directa del	(* Ecuaciones Simplificadas *)
	manipulador de tres grados de libertad, ya	
	que es la que contiene las ecuaciones va	$b1 = \{0, 0, L1\};$
	simplificadas. Puede observarse, que	b2 = {-4*A5*B2*L2, -2*A6*B2*L2, B1*L2};
	además de las ecuaciones de los eslabones	b3 = {-4*A5*C1*L3 -2*A6*C1*L3 C2*L3}
	y de la posición del elemento terminal se	
	y de la posición del elemento terminal, se	
	tiene una ecuación para e11 = e1; ésta sirve	(Finza)
	para mostrar el movimiento del primer	
	eslabón, que al girar sobre su propio eie.	b4 = {-4"A5"C1"L4, -2"A6"C1"L4, C2"L4};
	requiere que se dibuie una de las bases	b5 = {(A12*A6 - 4*A11*A5*C2)*L5,
	nernendiculares al él para ver claramente su	-2*(A12*A5 + A11*A6*C2)*L5, -4*A11*C1*L5};
	movimiente	
	novimiento.	(* Posición del elemento terminal *)
1	como al realizar la simulación es necesario	
	evaluar numericamente las ecuaciones del	$r_{n} = \{-4^{*}A5^{*}(B2^{*} 2 + C1^{*}(3+ 4))\}$
1	manipulador, se genera un respaido de	$-2^{*}\Delta 6^{*}(B2^{*} 2 + C1^{*}(13+ A))$
	dichas ecuaciones en esta misma sección,	= 2 A0 (D2 L2 + O1 (L3 + L4)), $= 1.4 \pm \text{D}4*(2 \pm \text{C}2*(1.2 \pm 1.4));$
	almacenándolas en nuevas variables.	L + D + L + C + C + (L + L +);
		(* Componente 1 de la Base local del eslabon 1
		(para dibujar la base) *)
		e11 = {-A6, 2*A5, 0};
		127
		(* Constantes de las ecuaciones *)
		A1 = P0*P1: A2 = 1-2*P0^2:
		$A3 = 00^{\circ}01$; $A4 = 1.2^{\circ}00^{\circ}2$;
1		$\Delta 5 = r10^{1}r13^{1}$ $\Delta 6 = 1.2^{1}r10^{2}$
		$A_0 = 110113$, $A_0 = 121102$, $A_7 = 2012010$, $A_8 = 4.222002$
		$A_{1} = 120121, A_{0} = 121202,$
		$A9 = r_{30}r_{31}$; $A10 = 1-2r_{30}r_{2}$;
		A11 = r40"r43; A12 = 1-2"r40^2;
		$B1 = A2^{*}A8 + 4^{*}A1^{*}A7; B2 = A2^{*}A7 + A1^{*}A8;$
		B3 = A10*A8 - 4*A7*A9; B4 = A10*A7 + A8*A9;
		B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4;
		C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6;
		(* Longitud de los estabones del manipulador *)
		(3 , , , , , , , , , , , , , , , ,
		L1 = 6: L2 = 5: L3 = 4: L4 = 2: L5 = 1.5:
		(* Respaldo de ecuaciones *)
		z1=h1: z2=h2: z3=h3: z4=h4: z5=h5: rx=rn:
		2 -0 , 22 -02, 20 -00, 27 -07, 20 -00, 1x -1p,
	Einalmente en esta narte del programa en	
	rinamente, en esta parte del programa, se	
	presenta la generación de las graticas que	
	nos permitiran observar la simulación de la	Graticación para visualizar la simulación.
	cinematica directa del manipulador en	
	cuestion.	CiearAii[r10,r13,r20,r21,r30,r31];
	Para lograr generar las gráficas adecuadas,	(* Se define el espacio tridimensional en que se
	para poder observar la simulación, primero	graficará el manipulador, que se usará como
	generamos mediante la función	gráfica base *)
	SurfaceGrafics una gráfica que contenda el	

```
espacio donde se mostrará el manipulador (* Suma de la longitudes de los eslabones 2.3 y 4 *)
con una superficie simulando el piso y con la
                                                           L234 = L2 + L3 + L4:
indicación de los ejes coordenados, la cual
se almacena inicialmente en las variables g1 g1=g2=SurfaceGraphics[{{0,0,0},{0,0,0},{0,0,0}},
y g2.
                                              MeshRange->{{-L234,L234}, {-L234,L234}},
                                              AxesLabel->{Eje X,Eje Y,Eje Z},
Definimos los parámetros de rotación de
                                                           DefaultFont->{"Negrita", 12},
cada eslabón, en función de los ángulos que
                                              PlotRange->{{-(L234+.1),(L234+.1)},
utilizaremos para girar cada uno de los
                                                           {-(L234+.1),(L234+.1)}.
eslabones, que serán:
                                                           {(L1-L234-.1),(L1+L234+.1)}},
                                              BoxRatios->{1,1,1},Axes->True]:
Eslabón 1 → theta1
                                            (* Se inicializan en cero los ángulos de rotación de
Eslabón 2 → theta2
Eslabón 3 → theta3
                                              los cuatro eslabones *)
Se generan dos ciclos For, el primero
                                               theta1 = 0; theta2 = 0; theta3 = 0; theta4 = 0;
(variable de control I) servirá para controlar
cual eslabón deberá moverse, o sea, cuándo (* Se definen los valores de los parámetros de las
                                              rotaciones *)
variar que ángulo, lo cual se logra utilizando
la función Switch, y el segundo ciclo
                                                         r10=Cos[theta1/2*Degree];
(variable de control ]) nos permitirá generar
varias gráficas del movimiento de cada
                                                         r13=Sin[theta1/2*Degree]:
                                                         r20=Cositheta2/2*Degreel:
eslabón. La variable I toma valores de 1 a 3.
ya que son tres eslabones y la variable }
                                                         r21=Sin[theta2/2*Degree];
toma valores de 1 a n, donde n es el número
                                                         r30=Cos[theta3/2*Degree]:
                                                         r31=Sin[theta3/2*Degree];
de gráficas a generar por cada eslabón.
                                                         r40=Cos[theta4/2*Degree]:
                                                         r43=Sin[theta4/2*Degree];
Una vez que los parámetros de rotación
tienen un valor numérico, se evaluan
                                       del (* Se inicializa un ciclo para mover los cuatro
numéricamente
                  las
                         ecuaciones
manipulador, haciendo uso del respaldo de
                                              eslabones, uno a la vez *)
ecuaciones realizado anteriormente
                                            Forfi=1, i<=4, i++,
Con las ecuaciones, ya numéricas, se
modifica la gráfica almacenada en g2 (* Se inicializa otro ciclo para generar varias gráficas
                                              para cada eslabón en diferentes posiciones*)
agregándole un punto en la posición que
corresponde a rp.
                                            For[j=1, j<=5, j++,
De la misma forma, con las ecuaciones
numéricas, se determinan los diferentes (* Se selecciona cual eslabón se moverá según el
                                              valor de la variable i, lo cual se logra cambiando
puntos de la gráfica a generar, es decir, se
                                              sus parámetros de rotación *)
genera un punto para cada inicio y final de
cada eslabón, además de generar puntos
para mostrar el vector e11 de la base del Switch[i,3, (* Se mueve primer eslabón *)
                                                        r10=N[Cos[theta1/2 Degree];
primer eslabón.
                                                        r13=N[Sin[theta1/2 Degree];theta1+=15,
Con los puntos anteriores, se genera la
                                                    2. (* Se mueve segundo eslabón *)
                                                        theta2+=15:
gráfica correspondiente a la posición del
                                                        r20=N[Cos[theta2/2 Degree];
manipulador, utilizando para ello la función
                                                        r21=N[Sin[theta2/2 Degree]; .
Graphics3D, que nos permite trazar líneas
                                                    1. (* Se mueve tercer eslabón *)
de diferente color y espesor entre los puntos
                                                        theta3+=15:
determinados.
                                                        r30=N[Cos[theta3/2 Degree];
                                                        r31=N[Sin[theta3/2 Degree]; ,
```

Finalmente se muestran en la pantalla, de	4, ("Se mueve cuarto eslabón ")
manera combinada, las gráficas g1, g2 y g3	theta4+=25;
que se generan en cada pasada de los	r40=NICositheta4/2 Degreel
cielos utilizando nara este la función Show la	rd2=bilSinitheted/2 Degree11
ciclos, utilizando para esto la función snow la	140-14[On[theta4/2 Degree]].
cual envia a la pantalla objetos gráficos y	
permite que se combinen varias gráficas.	(* Se asignan valores a las ecuaciones del
	manipulador, según los valores de los parámetros,
Se muestran a manera de ejemplo algunas	utilizando el respaldo de ecuaciones *)
or morestran, a manera de ejemple, algunde	
grancas de un proceso.	
	base=N[e11]; b1=N[z1]; b2=N[z2]; b3=N[z3];
	b4=N[z4]; b4=N[z4]; rp=N[rx];
	(* Longitud del vector que servirá de base del
	(Edighed der robier que der rid de base der
	manipulador)
	a≈3;
	(* Se definen los puntos en el espacio tridimensional
	para graficar el manipulador *)
	para grandar of manipulation /
	$punto1 = \{0, 0, 0\};$
}	punto2 = N[a*base];
	punto3 = N[-a*base];
	punto4 = N(b1)
	punto5 = N[punto4 + b2]
	puntos - Nountos + b2);
	puntoo = N[puntoo + Do],
	punto7 = N[punto6 + b5];
	punto8 = N[punto6 - b5];
	punto9 = N(punto7 + b4);
}	punto 10 = N[punto8 + b4]
	puno io - repanoo - ori,
	(* Se agrega a la granca g2 la posición del elemento
	terminal como un punto ")
	g2={Graphics3D[{PointSize[0.05],
	Point(rp], RGBColori0, 1, 01)1, g2);
	· · · · · · · · · · · · · · · · · · ·
	/t Co sessore la créfica del monipulador en la
	("Se genera la granca del manipulador en la
	posicion correspondiente ")
	g3=Graphics3D[{
	(* Base rotatoria *)
	(AbsoluteThickness/40) BGBColori1 3 01
	Line[{puntoz,punto3}]]},
	(* Eslabón 1 *)
	{AbsoluteThickness[25],RGBColor[0,1,0],
	Linel(punto1.punto4))).
	(* Felabón 2 *)
	(AbsoluteThickness/201 PCPColor(1, 4,0)
	Line[{punto4,punto5}]]},
	(* Eslabón 3 *)
	{AbsoluteThickness[15],RGBColor[0,0.5,1],
	Linel (punto5, punto6)]).



SIMULACION COMPUTACIONAL DE LA CINEMATICA



. . .

4.2.2 Cinemática Inversa.

La cinemática inversa del robot de cuatro grados de libertad, como se mencionó anteriormente, consiste en que, dada la posición y orientación del elemento terminal, se deben determinar los ángulos que deberán desplazarse cada una de las juntas del manipulador para alcanzar la posición deseada. En este caso, consiste en que dada la configuración no deformada (P₀, P₁, Q₀, Q₁) y la posición del elemento terminal ($r_p = \{r_px, r_py, r_pz\}$), se deben determinar los parámetros r4₀ y r4₃ que orienten el elemento terminal y los parámetros r1₀, r1₃, r2₀, r2₁, r3₀, r3₁ que satisfagan la ecuación de la posición del elemento terminal, que es la siguiente:

$$r_{p} = \{-4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}(L_{3}+L_{4})), \\ -2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}(L_{3}+L_{4})), L_{1} + B1^{*}L_{2} + C2^{*}(L_{3}+L_{4})\}$$
(4.20)

Para resolver la cinemática inversa del robot de cuatro grados de libertad, debemos plantear un sistema de 6 ecuaciones que nos permitan determinar los 6 parámetros de rotación que permitan ubicar el elemento terminal. De la ecuación de r_p obtenemos 3 ecuaciones y con las normas unitarias de los Quaterniones se completa el sistema, el cual se muestra a continuación.

$$r_{p}x = -4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}(L_{3}+L_{4}))$$

$$r_{p}y = -2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}(L_{3}+L_{4}))$$

$$r_{p}z = L_{1} + B1^{*}L_{2} + C2^{*}(L_{3}+L_{4}) \qquad (4.21)$$

$$r1_{0}^{2} + r1_{3}^{2} = 1$$

$$r2_{0}^{2} + r2_{1}^{2} = 1$$

$$r3_{0}^{2} + r3_{1}^{2} = 1$$

Siendo:

A1 =
$$P_0^*P_1$$
; A2 = 1-2* P_0^2 ;
A3 = $Q_0^*Q_1$; A4 = 1-2* Q_0^2 ;
A5 = $r_1_0^*r_1_3$; A6 = 1-2* $r_1_0^2$;
A7 = $r_2_0^*r_2_1$; A8 = 1-2* $r_2_0^2$;
A9 = $r_3_0^*r_3_1$; A10 = 1-2* $r_3_0^2$; (4.22)
B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8;
B3 = A10*A8 - 4*A1*A7; B4 = A10*A7 + A8*A9;
B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4;
C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6;

Lo anterior nos permite ubicar el elemento terminal en el punto { r_px , r_py , r_pz } dentro de su espacio de trabajo, ahora si deseamos que el elemento terminal siga una trayectoria determinada, una de las cosas que podemos hacer es discretizar dicha trayectoria en varios puntos a lo largo de ella y determinar los parámetros de rotación de los eslabones del manipulador para cada punto, resolviendo el sistema de ecuaciones (4.21), lo que nos permitirá ubicar el elemento terminal sobre dicha secuencia de puntos. Lo anterior no permite ublcar el elemento terminal en cualquier punto de la trayectoria, solo en los puntos en que dicha trayectoria se discretizó.

Para determinar la orientación del elemento terminal (r4₀ y r4₃), debemos considerar el hecho de que, cuando la pinza esta en la posición inicial (b₅ paralelo al eje x), su orientación con respecto al plano de la base del manipulador (plano x-y) no cambia con los movimientos del los eslabones. Sabiendo lo anterior, podemos orientar la pinza con respecto al plano x-y, en cualquier

posición de los eslabones, con una rotación fija, pero si se desea orientarlo con respecto a otra referencia, deberá determinarse el ángulo formado entre el vector b₅ y una línea perpendicular a b₄. A continuación se presenta el programa desarrollado en Mathematica[®] para la cinemática inversa del manipulador de cuatro grados de libertad, el cual es continuación del anterior donde se presenta la cinemática directa.

Este programa es parte integral del anterior, es	
decir, para que funcione deberá tenerse la	
parte anterior donde se presenta modelación y	CINEMATICA INVERSA
la cinematica directa, ya que se requieren las	
ecuaciones del manipulador. En esta seccion	Solucion de ecuaciones para diferentes
se presenta el carculo para tres diferentes	trayectorias
cheen/ar la simulación	
En este nerte se determinen les nertenetres de	LINEA DECTA
En esta parte se determinan los parametros de	LINEA REGIA
de puntos que pertenecen a una línea recta la	/* Se inicializes variables *)
cual se define por sus dos puntos extremos	
(S0 S1) Para lo anterior primeramente	"eoir?" "eoir?" "eoir?" "eoircion"):
recuperamos las ecuaciones del manipulador	
las cuales se respaldaron en otras variables	(*Se recuperan las ecuaciones del manipulador*)
(deberá resolverse la cinemática inversa nara	h1=z1: b2=z2: b3=z3: b4=z4: b5=z5: rn=ry:
determinar la configuración no deformada v	rn1=rn[[1]]; rn2=rn[[2]]; rn3=rn[[3]];
ejecutar la celda donde están las ecuaciones	
optimizadas para la configuración deformada).	(* Variables de control *)
	t = 0: (* Para discretizar en varios puntos la
Para determinar la puntos sobre la travectoria.	travectoria seleccionada *)
utilizamos la forma paramétrica de una recta	i = 1; (* Para guardar la solución de cada punto
que pasa por dos puntos $p_1 = \{x_1, y_1, z_1\}$ y	de la trayectoria *)
$p_2 = \{x_1, y_1, z_1\}$	
	(* Valores iniciales para resolver el sistema de
$px = x_1 + (x_2 - x_1)t$	ecuaciones *)
$py = y_1 + (y_2 - y_1)t$	
$pz = z_1 + (z_2 - z_1)t$	ir10=.7071; ir13=.7071; ir20=.7071; ir21=.7071;
	ir30=.7071; ir31=.7071;
Con lo anterior se realiza un cicio utilizando	/* Travastaria en llasa sasta dada por las nuntas
como vanable de control del mismo, a la	(Trayectoria en línea recta dada por los puntos
cobre la travectoria a cequir y resolvemos el	extremes)
eistema de ecuaciones (4.21)	S0=/-8-610\·S1=/8-64\·
r_x = - 4*A5*(B2*L, + C1*L,),	(* Se inicia un ciclo para resolver el problema
	cinemático inverso en los puntos en que será
$r_p y = -2$ AG (BZ $L_2 + Cr L_3)$,	discretizada la trayectoria *)
$r_p z = L_1 + B1^*L_2 + C2^*L_3$	Fod-0 441 44-1
$r1_0^2 + r1_3^2 = 1$	Por[t=0, t < 1, t = .1, t = .1]
$r2_0^2 + r2_1^2 = 1$	(* Se determinan las coordenadas del punto
$r3_0^2 + r3_1^2 = 1$	correspondiente de la trayectoria")
Este sistema de estraismen es resurbis	px=SU[[1]]+(S1[[1]]-SU[[1]])*(;
Este sistema de ecuaciones se resueive	py=00[[2]]+(01[[2]]-00[[2]])*(
utilizando la funcion Findroot Usando los	pz=ou[[o]]+(o i[[o]]-ou[[o]]) t,

valores iniciales mostrados. Una vez que se resuelve el sistema de ecuaciones anterior, se verifica si la precisión de la solución es la adecuada si es así se fijan como valores	(* Solución del problema cinemático inverso para el punto seleccionado *) solucion[i] =
iniciales, para resolver la cinemática inversa del siguiente punto, a la solución obtenida y se continua el ciclo, si la solución está fuera de la precisión se envía un mensaie y se restauran	FindRoot[{rp1 == px, rp2 == py, rp3 == pz, r10^2+r13^2 == 1, r20^2+r21^2 == 1, r30^2+r31^2 == 1}, (r10 ir10) (r13 ir13) (r20 ir20)
los valores iniciales continuando el ciclo hasta terminar. Una vez determinados los parámetros para	{r21,ir21},{r30,ir30},{r31,ir31}, MaxIterations->50];
ubicar el elemento terminal sobre la trayectoria, se determina la rotación necesaria para su orientación en cada punto de la	(* Se verifica la precisión de la solución *) e = 10^-5; (* Precisión requerida *)
trayectoria (r4 ₀ , r4 ₃). En este caso deseamos que el eje transversal de la pinza (b ₅) sea perpendicular a la trayectoria propuesta. Para determinar los parámetros r4 ₀ y r4 ₃ , calculamos	If[Abs[px-(rp1/.solucion[i])] < e && Abs[py-(rp2/.solucion[i])]< e && Abs[pz-(rp3/.solucion[i])] < e,
el ángulo formado entre una línea que sea perpendicular a la trayectoria y perpendicular a b_4 (producto cruz entre la trayectoria y b_4) y el eje transversal de la pinza b_5 .	(* Si la solución está en la precisión fijada, se definen como valores iniciales, para resolver la cinemática inversa del siguiente punto de la trayectoria a los valores de la solución obtenida, si no es así se restauran los valores iniciales utilizados *)
	ir10=r10/.solucion[i]; ir13=r13/.solucion[i]; ir20=r20/.solucion[i]; ir21=r21/.solucion[i]; ir30=r30/.solucion[i]; ir31=r31/.solucion[i],
	(* Se imprime un mensaje si la solución no está en la precisión definida *)
	Print["No se encontró solución para el punto:", {px,py,pz}];
	(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
	ir10=.7071; ir13=.7071; ir20=.7071; ir21=.7071; ir30=.7071; ir31=.7071];
	(* Se incremente la variable i y se cierra el ciclo For *)
	i++]
	(* Se determina la orientación de la pinza (perpendicular a la trayectoria) *)
	(* Se almacenan en variables separadas la solución para cada par de parámetros *)

	For[j=1, j <i, j++,="" solr1[j]="{r10,r13}/.solucion[j];<br">solr2[j]={r20,r21}/.solucion[j]; solr3[j]={r30,r31}/.solucion[j]];</i,>
	(* Se determina un vector unitario en dirección de la trayectoria (tray) *)
	tray = (S0 - S1); tray = (tray)/(Sqrt[tray[[1]]^2 + tray[[2]]^2 + tray[[3]]^2]);
	(* Se inicia un ciclo para determinar la orientación de la pinza en cada punto de la trayectoria *)
	For[j=1, j <i, j++,<="" td=""></i,>
	(* Se asigna el valor de los parámetros para el punto de la trayectoria correspondiente almacenados en las variables solr?[j] *)
	r10=solr1[j][[1]];r13=solr1[j][[2]]; r20=solr2[j][[1]];r21=solr2[j][[2]]; r30=solr3[j][[1]];r31=solr3[j][[2]];
	(* Se ubica la pinza en la posición original *) r40=1;r43=0;
	(* Se determina un vector perpendicular a la trayectoria y perpendicular al eje longitudinal de la pinza (b4) *)
	Lh = Cruz[b4,tray];
	(* Se calcula el ángulo formado entre el vector anterior y el eje transversal de la pinza (b5), que es el ángulo que debe rotar la pinza *)
	cosenoang = (b5.Lh)/(Sqrt[Lh[[1]]^2 + Lh[[2]]^2 + Lh[[3]]^2]*L5);
	lf[cosenoang!=0, ang = ArcCos[cosenoang], ang =90*Degree] ;
	(* Se almacena en forma de parámetros el ángulo que debe rotar la pinza *)
	solr4[j] = {Cos[ang/2],Sin[ang/2]};
En esta sección, al igual que en la anterior se	CIRCUNFERENCIAS O ELIPSES
secuencia de puntos sobre una trayectoria,	(* Se inicializan variables *)
que en este caso se trata de una elipse o circunferencia. El proceso que se sigue es el	ClearAll[r10,r13,r20,r21,r30,r31,r40,r43,"solr1", "solr2","solr3","solr4", "solucion"];
mismo que para la línea recta, solamente	· · ·



	(* Si la solución está en la precisión fijada se definen como valores iniciales, para resolver la cinemática inversa del siguiente punto de la trayectoria, a los valores de la solución obtenida, si no es así se restauran los valores iniciales utilizados *)
	ir10=r10/.solucion[i]; ir13=r13/.solucion[i]; ir20=r20/.solucion[i]; ir21=r21/.solucion[i]; ir30=r30/.solucion[i]; ir31=r31/.solucion[i],
	(* Se imprime un mensaje si la solución no está en la precisión definida *)
	Print["No se encontró solución para el punto:", {px,py,pz}];
	(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
	ir10=.7071; ir13=.7071; ir20=.3; ir21=.9; ir30=.5; ir31=.8];
	(* Se incremente la variable i y se cierra el ciclo For *)
	i++]
	(* Se determina la orientación de la pinza, se considera que el eje transversal de la pinza deberá se paralelo al plano x-y *)
	(* Como el eje transversal de la pinza no cambia su orientación al moverse el manipulador, si arranca de su posición inicial (b5 paralelo al plano x-y), simplemente la pinza debe rotar cero grados *)
	For[j=1, j <i, j++,="" solr4[j]="{Cos[0/2],Sin[0/2]}];</th"></i,>
De la misma forma que se determinan los parámetros para las travectorias anteriores.	ASTROIDE
ahora se determinará una trayectoria en forma de astroride , la cual se ubicará en el plano x-z, con centro en {h, k, l} cuya forma	(* Se inicializan variables *) ClearAll[r10,r13,r20,r21,r30,r31,r40,r43,"solr1", "solr2","solr3","solr4", "solucion"];
paramétrica es: px = radio*Cos ³ (t) + h; py = k; pz = radio*Sin ³ (t) + l;	(*Se recuperan las ecuaciones del manipulador*) b1=z1; b2=z2; b3=z3; b4=z4; b5=z5; rp=rx; rp1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]];
	(* Variables de control *) t = 0; (* Para discretizar en varios puntos la trayectoria seleccionada *)

```
i = 1; (* Para guardar la solución de cada punto
                                                   de la trayectoria *)
                                           (* Variable de control para definir la travectoria *)
                             {h, k, l}
                                                   radio = 6: (* Radio de la astroide *)
                                                   h=0;k=-5;l=7; (* Centro de la elipse *)
                                           (* Valores iniciales para resolver el sistema de
                                             ecuaciones *)
                       radio
                                           ir10 = .7071; ir13 = .7071; ir20 = .3; ir21 = .9;
radio
                                           ir30 = .5; ir31 = .8;
                                           (* Se inicia un ciclo para resolver el problema
                                             cinemático inverso en los puntos en que será
                                             discretizada la trayectoria *)
                                           For[t=0, t<=2*N[Pi], t+=N[Pi]/11,
                                           (* Se determinan las coordenadas del punto
                                             correspondiente de la travectoria *)
                                                     px = radio*Cos[t]^3 + h;
                                                     py = k;
                                                     pz = radio*Sin[t]^3 + I;
                                           (* Solución del problema cinemático inverso para
                                             el punto seleccionado *)
                                           solucion[i] =
                                           FindRoot[{rp1 == px, rp2 == py, rp3 == pz,
                                                     r10^2+r13^2 == 1, r20^2+r21^2 == 1,
                                                     r30^2+r31^2 == 1},
                                                     {r10,ir10},{r13,ir13},{r20,ir20},
                                                     {r21,ir21},{r30,ir30},{r31,ir31},
                                                     MaxIterations->50];
                                           (* Se verifica la precisión de la solución *)
                                           e = 10^-5; (* precisión requerida *)
                                           If[Abs[px-(rp1/.solucion[i])] < e &&
                                             Abs[py-(rp2/.solucion[i])] < e &&
                                             Abs[pz-(rp3/.solucion[i])] < e,
                                           (* Si la solución está en la precisión fijada se
                                             definen como valores iniciales, para resolver la
                                             cinemática inversa del siguiente punto de la
                                             trayectoria, a los valores de la solución obtenida,
                                             si no es así se restauran los valores iniciales
                                             utilizados *)
                                           ir10=r10/.solucion[i]; ir13=r13/.solucion[i];
                                           ir20=r20/.solucion[i]; ir21=r21/.solucion[i];
```

	ir30=r30/.solucion[i]; ir31=r31/.solucion[i],
	Sector of Astronomy March 2012
	(* Se imprime un mensaje si la solución no está en la precisión definida *)
	Print["No se encontró solución para el punto: ", {px,py,pz}];
	(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
	ir10=.7071; ir13≃.7071; ir20=.3; ir21=.9; ir30≃.5; ir31≃.8];
	(* Se incremente la variable i y se cierra el ciclo For *)
	i++]
	(* Se determina la orientación de la pinza, se considera que el eje transversal de la pinza deberá se paralelo al plano x-y *)
	(* Como el eje transversal de la pinza no cambia su orientación al moverse el manipulador si arranca de su posición inicial (b5 paralelo al plano x-y), simplemente la pinza debe rotar cero grados *)
	For[j=1, j <i, j++,="" solr4[j]="{Cos[0/2],Sin[0/2]}];</th"></i,>
Una vez determinados los parámetros de rotación para la trayectoria seleccionada, abora se generarán las gráficas que permitirán	Graficación para visualizar la simulación de la trayectoria seleccionada.
ver la simulación de la cinemática inversa del	ClearAll[r10,r13,r20,r21,r30,r31,r40,r43]
poder realizarlo, recuperamos los valores de los parámetros almacenados en la variable solución[j] y los almacenamos en variables separadas para cada par de parámetros. Al igual que en la cinemática inversa, se genera	(* Se recuperan los parámetros de rotación almacenados en la variable solucion[j], obtenidos de resolver el problema cinemático inverso y se almacenan en variables para cada par de parámetros *)
graficar el manipulador, la cual se almacena en las dos variables g1 y g2, posteriormente se inicia un ciclo, que se repetirá tantas veces	For[j=1, j <i, j++,="" solr1[j]="{r10,r13}/.solucion[j];<br">solr2[j]={r20,r21}/.solucion[j]; solr3[i]={r30,r31}/.solucion[i]];</i,>
como valores de los parámetros se hayan calculado para la trayectoria seleccionada (variable "í"). En cada pasada del ciclo se asignan los valores correspondientes a los parámetros y con ellos se capara la gráfica del	(* Se define el espacio tridimensional en que se graficará el manipulador, que se usará como gráfica base *)
manipulador de la misma forma que se hizo en la cinemática directa. Se mostrarán, a manera	(*Suma de la longitudes de los eslabones 2,3 y 4 *) L234 = L2 + L3 + L4;

de ejemplo, trayectoria.	algunas	gráficas	de	una	<pre>g1=g2=SurfaceGraphics[{{0.0.0}.{0.0.0}.{0.0.0}}, MeshRange->{{-L234,L234}.{-L234,L234}}, AxesLabel->{Eje X, Eje Y, Eje Z}, DefaultFont->{"Negrita", 12}, PlotRange->{{-(L234+.1),(L234+.1)}, {-(L234+.1),(L234+.1)}, {(L1-L2341),(L1+L234+.1)}}, BoxRatios->{1,1,1},Axes->True]; (* Se inicia un ciclo para generar las gráficas de cada punto de la trayectoria ") For[j=1, j<i, j++,<br="">(* Se asigna el valor de los parámetros para el punto de la trayectoria correspondiente almacenados en las variables solr?[j]*) r10=solr1[j][[1]];r13=solr1[j][[2]]; r20=solr2[j][[1]];r21=solr2[j][[2]]; r30=solr3[j][[1]];r31=solr3[j][[2]]; r40=solr4[j][[1]];r43=solr4[j][[2]]; r40=solr4[j][[1]];r43=solr4[j][[2]]; de solr4[j][[1]];r43=solr4[j][[2]]; r40=solr4[j][[1]];r4]=solr4[j][[2]]; r40=solr4[j][[1]];r4]=solr4[j][[2]]; r40=solr4[j];b1=N[z1];b2=N[z2];b3=N[z3]; b4=N[z4];b5=N[z5];rp=N[rx]; (* Longitud del vector que servirá de base del manipulador *) a = 3;</i,></pre>
			(* Longitud del vector que servirá de base del manipulador *)		
					a = 3;
			(* Se definen los puntos en el espacio tridimensional para graficar el manipulador *)		
		punto1 = {0,0,0}; punto2 = N[a*base]; punto3 = N[-a*base];			
					punto5 = N[punto4 + b2];
		punto6 = N[punto5 + b3];			
		punto r = N[puntos + b5];			
			punto9 = N[punto7 + b4]:		
					punto10 = N[punto8 + b4];
		(* Se agrega a la gráfica base (g2) la posición del elemento terminal, como un punto *)			
		g2 = {Graphics3D[{PointSize[0.01], Point[rp],RGBColor[0.1.0]}],g2};			



SIMULACION COMPUTACIONAL DE LA CINEMATICA



4.3 Cinemática de un robot tipo SCARA de cuatro grados de libertad.

En la siguiente figura se muestra la configuración del robot que se simulará en está sección, el cual fue previamente modelado en el capítulo anterior.



Figura 4.3 Configuración del manipulador SCARA de cuatro grados de libertad

4.3.1 Cinemática Directa.

El problema cinemático directo para este manipulador, consiste en que dados los desplazamientos angulares θ_1 , θ_2 , θ_4 , y el desplazamiento lineal L_{3v} , se debe determinar la posición y orientación del elemento terminal, así como la configuración resultante de sus eslabones. Para realizar lo anterior hacemos uso de las ecuaciones (3.108), (3.109), (3.110) determinadas en el capitulo anterior, que al evaluarlas se resuelve la cinemática directa de este manipulador. Dichas ecuaciones se muestran a continuación.

$$b_{1} = \{B1^{*}L_{1}, -2^{*}B2^{*}L_{1}, 0\}$$

$$b_{2} = \{C1^{*}L_{2}, -2^{*}C2^{*}L_{2}, 0\}$$

$$b_{3} = \{0, 0, -L_{3}\}$$

$$b_{4} = \{0, 0, -(L_{3}+I_{3})\}$$

$$b_{4} = \{0, 0, -L_{4}\}$$

$$b_{5} = \{(4^{*}A9^{*}C2 - A10^{*}C1)^{*}L_{5}, 2^{*}(A9^{*}C1 + A10^{*}C2)^{*}L_{5}, 0\}$$

$$r_{0} = \{B1^{*}L_{1} + C1^{*}L_{2}, -2^{*}(B2^{*}L_{1} + C2^{*}L_{2}), -(L_{3} + L_{3}) + L_{4}\}$$

$$(4.23)$$

Donde:

A1 =
$$P_0^*P_3$$
; A2 = 1-2* P_0^2 ;
A3 = $Q_0^*Q_3$; A4 = 1-2* Q_0^2 ;

Las variables de las ecuaciones anteriores tienen el siguiente significado:

L₁, L₂, L₃, L₃, L₄, L₅ \rightarrow Longitud de los eslabones del manipulador. b₁, b₂, b₃, b₃, b₄, b₅ \rightarrow Eslabones del manipulador. r_p \rightarrow Posición del elemento terminal. P₀, P₃, Q₀, Q₃ \rightarrow Parámetros de rotación en la configuración no deformada. r1₀, r1₃, r2₀, r2₃, r4₀, r4₃ \rightarrow Parámetros de rotación en la configuración deformada.

Los parámetros de rotación tienen la siguiente relación con las rotaciones de los eslabones del manipulador.

$$\begin{array}{ll} P_{0} = \cos \left(\phi_{1}/2 \right); & P_{3} = \sin \left(\phi_{1}/2 \right); \\ Q_{0} = \cos \left(\phi_{2}/2 \right); & Q_{3} = \sin \left(\phi_{2}/2 \right); \\ r1_{0} = \cos \left(\theta_{1}/2 \right); & r1_{3} = \sin \left(\theta_{1}/2 \right); \\ r2_{0} = \cos \left(\theta_{2}/2 \right); & r2_{3} = \sin \left(\theta_{2}/2 \right); \\ r4_{0} = \cos \left(\theta_{4}/2 \right); & r4_{3} = \sin \left(\theta_{4}/2 \right); \\ \end{array}$$

$$\begin{array}{l} (4.26) \\ (4.27) \\ (4.27) \end{array}$$

Siendo:

φ₁ el ángulo de rotación del eslabón 2 en la configuración no deformada.
φ₂ el ángulo de rotación del eslabón 3 en la configuración no deformada.

θ1 el ángulo de rotación del eslabón 1 en la configuración deformada.

θ2 el ángulo de rotación del eslabón 2 en la configuración deformada.

θ, el ángulo de rotación de la pinza en la configuración deformada.

Antes de poder realizar el proceso de cinemática directa del manipulador, debemos determinar la configuración no deformada del robot mediante un proceso de cinemática inversa, que consiste en calcular los parámetros $P=\{P_0,P_1,0,0\}$ y $Q=\{Q_0,Q_1,0,0\}$ dada la posición del elemento terminal R_p , que en este caso es el extremo del eslabón 3 variable donde esta unida la pinza (Ecuaciones (3.96), (3.97)). Es decir, con las siguientes ecuaciones puede plantearse un sistema de ecuaciones, como se muestra.

$$\mathbf{R}_{p} = \{-\mathbf{A2^{*}L_{1}} + \mathbf{L_{2}^{*}}(\mathbf{A2^{*}A4} - \mathbf{4^{*}A1^{*}A3}), \mathbf{2^{*}}(\mathbf{A1^{*}L_{1}} - \mathbf{L_{2}^{*}}(\mathbf{A2^{*}A3} + \mathbf{A1^{*}A4})), \mathbf{-}(\mathbf{L_{3}^{+}I_{3^{*}}})\}; \quad (4.28)$$

Donde:

A1 =
$$P_0^*P_1$$
; A2 = 1 - 2* P_0^2 ; (4.29)
A3 = $Q_0^*Q_1$; A4 = 1 - 2* Q_0^2 ;

Con la ecuación para R_p podemos plantear 3 ecuaciones si damos la posición para el elemento terminal como $R_p = \{R_px, R_py, R_pz\}$ y con las normas unitarias de los Quaterniones P y Q completamos un sistema de 5 ecuaciones con 5 incógnitas, que al resolverlo obtenemos los parámetros que determinan la configuración no deformada.

(4.25)

$$-A2^{*}L_{1}+L_{2}^{*}(A2^{*}A4-4^{*}A1^{*}A3) = R_{p}x$$

$$2^{*}(A1^{*}L_{1}-L_{2}^{*}(A2^{*}A3 + A1^{*}A4)) = R_{p}y$$

$$-(L_{3}+I_{3}) = R_{p}z \qquad (4.30)$$

$$P_{0}^{2} + P_{1}^{2} = 1$$

$$Q_{0}^{2} + Q_{1}^{2} = 1$$

Con lo anterior podemos ahora resolver el problema cinemático directo y realizar la correspondiente simulación computacional, que como se mencionó se realizará en Mathematica[®]. A continuación se presenta el programa desarrollado para este caso. Con el objeto de mostrar la forma de utilizar Mathematica[®] para obtener las ecuaciones del manipulador, tanto en su configuración no deformada como en la deformada.

En la primera parte del programa se define la	Definición de la Transformación Rotación
transformación rotación de un vester	
transionnación rotación de un vector	
tridimensional y de un Quaternion, y de	(" Se apagan algunos mensajes del sistema ")
algunas otras funciones necesarias que se	
muestran a continuación.	Off[General::speli1]
	Off[General::spell]
Transformación de un Quaternión en un	
vector tridimensional	/* Se define la Transformación de un Ousternión e
votor thannonaionai.	up vester tridimagional Tul) *)
$q = \{q_0, q_1, q_2, q_3\} \in Q$	
	Tv[q_]:={q[[2]],q[[3]],q[[4]]};
$Tv[q] = {q_1, q_2, q_3}.$	
	(* Se define la Transformación inversa de Tv[] *)
Transformación vectorial inversa	Tvinvly_1;={0,v[[1]],v[[2]],v[[3]]);
$y = fy + y + y = 0^3$	/* Se define la Transformación de un Ousternión a
$v = \{v_1, v_2, v_3\} \in \mathcal{D}^*$	
	Quaternion vectorial -)
$Tvinv[v] = \{0, v_1, v_2, v_3\}.$	
	TvQ[q_]:={0,q[[2]],q[[3]],q[[4]]};
Transformación de un Quaternión en un	
Quaternión vectorial.	(* Se define la Multiplicación entre dos Quaterniones
	•)
$T_{V}O[\alpha] = \{0, \alpha, \alpha\}$,
Producto de dos Quaterniones.	
	p[[1]]*q[[2]] + p[[2]]*q[[1]] + p[[3]]*q[[4]] - p[[4]]*q[[3]],
$p = \{p_0, p_1, p_2, p_3\}, q = \{q_0, q_1, q_2, q_3\} \in Q$	p[[1]]*q[[3]] - p[[2]]*q[[4]] + p[[3]]*q[[1]] + p[[4]]*q[[2]],
	p[[1]]*q[[4]] + p[[2]]*q[[3]] - p[[3]]*q[[2]] + p[[4]]*q[[1]]);
ProductoQ[p.g] =	
{D*"0** D*"0** D*"0** D*"0**	(* Se define el conjugado de un Quaternión *)
P0 41 + P1 40 + P2 43 - P3 42;	ConjugadoQin_1;=/n[[1]] -n[[2]] -n[[3]] -n[[4]]);
Po 42 * P1 43 * P2 40 * P3 41	
Po ⁻ q ₃ + P ₁ ⁻ q ₂ - P ₂ ⁻ q ₁ + P ₃ ⁻ q ₀ }	/t Ca define la Transformación Batación de un
Conjugado de un Quaternión.	vector tridimensional *)

ConjugadoQ[p] = { $p_0, -p_1, -p_2, -p_3$ }	Rotacion[p_,v_]:≖ Tv[ProductoQ[p, ProductoQ[TvInv[v],
Rotación de un vector v $\in \Re^3$ con un Quaternión p.	ConjugadoQ[p]]]];
- r	(* Se define la Transformación Rotación de un
Rotacion[p,v] = Tv[p*Tvinv[v]* p]	Quaternión *)
Rotación de q e Q con un Quaternión p.	RotacionQ[p_,q_]:={q[[1]],0,0,0}+
considerando que el primer elemento no	TvQ[ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]];
cambia al rotar cuando pi = 1 .	(* Se define el Producto Cruz entre dos vectores
RotacionQ[p,q] = { $q_0,0,0,0$ } + TvQ[$p^*q^*\bar{p}$]	tridimensionales *)
	Cruz[a_,b_]:= {a[[2]] b[[3]} - a[[3]] b[[2]],
	- a [[1]] b[[3]] + a [[3]] b[[1]],
	a[[1]] b[[2]] - a[[2]] b[[1]]}
Se realizan definiciones iniciales necesarias, como son la base canónica de 93º y las bases	Definiciones iniciales
que servirán como bases locales en los	(* Base canónica *)
eslabones. También se define la forma local de los Quaterniones que nos permitirán rotar, tanto en la posición no deformada como en la deformada.	e1 = {1,0,0}; e2 = {0,1,0}; e3 = {0,0,1};
	(* Bases locales de los eslabones *)
	(* Eslabón 1 *)
	e1 = {1,0,0}; e2 = {0,1,0}; e3 = {0,0,1};
	(* Eslabón 2 *) f1 = {1,0,0}; f2 = {0,1,0}; f3 = {0,0,1};
	(* Eslabón 3 *)
	ן g1 = {1,0,0}; g2 = {0,1,0}; g3 = {0,0,1};
	(* Esiabón 4 (pinza) *) h1 = {1,0,0}; h2 = {0,1,0}; h3 = {0,0,1};
	(*Quaterniones para rotar a posición no deformada*)
	P = {P0,0,0,P3}; (* Eslabón 2 *)
	Q = {Q0,0,0,Q3}; (* Eslabón 3 *)
	(* Quaterniones para rotar a posición deformada *)
	r1 = {r10, 0, 0, r13}; (* Eslabón 1 *)
	r2 = {r20, 0, 0, r23}; (* Eslabón 2 *)
	r4 = {r40, 0, 0, r43}; (* Eslabon 4 *)
En esta sección del programa se realiza la	MODELADO DE LA CONFIGURACION NO
Modelación de la configuración no deformada	DEFORMADA
dei manipulador. Como la forma de las bases locales de los eslabones, en la configuración	Modelo para la configuración no deformada.
no deformada, se obtuvo en el capitulo	

anterior, entonces para facilitar la obtención del las ecuaciones del manipulador.	(* Se inicializan variables *) ClearAll[P0,P1,Q0,Q1,L1,L2,L3,b1,b2,b3];
forma de las bases locales, que son:	(* Se definen variables para las normas unitarias de los Quaterniones P y Q *)
$\mathbf{e}_{j}^{i} = \rho(\mathbf{P}, \mathbf{e}_{j})$ (Eslabon 1)	
$\mathbf{f}_{j}^{II} = p((\mathbf{P}^{*}\mathbf{Q}), \mathbf{f}_{j}) (\text{Eslabon 2})$	normas1= { $P1->(1-P0^2)^{(1/2)},Q1->(1-Q0^2)^{(1/2)}$; normas2= { $(1-P0^2)^{(1/2)}->P1,(1-Q0^2)^{(1/2)}->Q1$;
$\mathbf{g}_{j}^{H} = \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{g}_{j})$ (Eslabon 3)	(* Se obtienen los productos de Quaterniones
$\mathbf{h}_{j}^{II} = \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{h}_{j})$ (Pinza)	necesarios para rotar las bases *)
Para lo anterior es necesario definir las	PQ = ProductoQ[P,Q];
normas unitarias de los Quaterniones P y Q. las cuales se almacenan en las variables "normas1" y "normas2", para aplicarlas en las	(* Se obtiene la forma de las bases locales de los eslabones en la configuración no deformada *)
ecuaciones resultantes de las rotaciones.	(* Esiabón 1 *)
Las variables que se utilizan para las bases	e11 = Expand[Expand[Rotacion[P,e1]]//.normas1]
locales de los eslabones son:	e21 = Expand[Expand[Rotacion[P,e2]]//.normas1]
ej = Primer eslabón. fi = Segundo eslabón	e31 = Expand[Expand[Rotacion[P,e3]]//.normas1]
gj = Tercer eslabón.	//.normas2;
hj = Pinza. j = 1,2,3.	(* Esiabón 2 *)
	f12 = Expand[Expand[Rotacion[PQ,f1]]//.normas1] // normas2
	f22 = Expand[Expand[Rotacion[PQ,f2]]//.normas1]
	f32 = Expand[Expand[Rotacion[PQ,f3]]//.normas1]
	//.normas2;
	(* Esiabón 3 *)
	y 12 = Expand(Expand(Rotacion(PQ,g1))//.normas1] //.normas2;
	g22 = Expand[Expand[Rotacion[PQ,g2]]//.normas1] //.normas2;
	g32 = Expand[Expand[Rotacion[PQ.g3]]//.normas1] //.normas2;
	(* Eslabón 4 *)
	h12 = Expand[Expand[Rotacion[PQ,h1]]//.normas1] //.normas2:
	h22 = Expand[Expand[Rotacion[PQ,h2]]//.normas1] // normas2
	h32 = Expand[Expand[Rotacion[PQ,h3]]//.normas1] //.normas2;
Los eslabones del manipulador y la posición	(* Vectores que definen los eslabones del
del elemento terminal son identificados, en la posición no deformada por las variables	manipulador *)
siguientes:	b1 = L1*e11; (* Eslabón 1 *) b2 = L2*f12; (* Eslabón 2 *)

b1 = Eslabón 1. b3 = L3*(-g32); (* Eslabón 3 fijo *) b2 = Eslabón 2. b3v = I3v*(-q32); (* Eslabón 3 variable *) b3 = Eslabón 3. b4 = L4*(-h32); (* Pinza *) b4 = Eslabón 3. b5 = L5*h12: (* Pinza *) b5 = Eslabón 3. (* Se simplifican las expresiones *) Rp = Posición del elemento terminal. constantes = {P0*P3->A1, P0^2->(1-A2)/2, Las ecuaciones se simplifican utilizando las Q0"Q3->A3, Q0^2->(1-A4)/2}; siguientes constantes.. b1 = Expand[Expand[b1//.constantes]//.constantes]; A1 = P0*P1: A2 = 1 - 2*P0^2: b2 = Expand[Expand[b2//.constantes]//.constantes]; $A3 = Q0^{\circ}Q1$; $A4 = 1 - 2^{\circ}Q0^{\circ}2$; b5 = Expand[Expand[b5//.constantes]//.constantes]; (* Vector que define posición del elemento terminal*) Rp = b1+b2+b3+b3v;Es esta parte se muestran las ecuaciones Ecuaciones optimizadas para la posición no deformada. obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P, Q y I3v, que definen la configuración no ClearAll[P0,P3,Q0,Q3,I3v] deformada en una posición dada del elemento terminal, que será de donde (* Ecuaciones optimizadas *) arrangue la simulación de la cinemática directa o inversa del manipulador en b1 = {-A2*L1, 2*A1*L1, 0}; (* Eslabón 1 *) cuestión. Es decir, antes de generar una $b2 = {(A2^{*}A4 - 4^{*}A1^{*}A3)^{*}L2,$ simulación computacional de la cinemática - 2*L2*(A2*A3 + A1*A4), 0}; (* Eslabón 2 *) directa o inversa del manipulador, debemos determinar la configuración no deformada $b3 = \{0, 0, -L3\}$ (* Eslabón 3 filo *) realizando un proceso de cinemática inversa $b3v = \{0, 0, -|3v\};$ (* Eslabón 3 variable *) para calcular P y Q para una posición conocida del elemento terminal, la cual debe (* pinza *) tener la forma siguiente: $b4 = \{0, 0, -L4\}$ b5 = {(A2*A4 - 4*A1*A3)*L5. $Rp = \{Rpx, Rpy, Rpz\}.$ - 2*L5*(A2*A3 + A1*A4), 0); (* Posición del elemento terminal *) $R_{D} = \{L2^{*}(A2^{*}A4 - 4^{*}A1^{*}A3) - A2^{*}L1, ..., A2^{*}L1\}$ 2*(A1*L1-L2*(A2*A3+A1*A4)), -(L3+I3v)}; (* Constantes de las ecuaciones *) A1 = P0*P3; A2 = 1 - 2*P0^2; $A3 = Q0^{\circ}Q3;$ $A4 = 1 - 2^{\circ}Q0^{\circ}2;$ (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 1;

Se resuelve el sistema de ecuaciones mostrado en (4.19), para P0,P1,Q0,Q1, utilizando la función FindRoot cuya sintaxis	Cinemática Inversa para determinar la Configuración no Deformada
se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo.	(* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" y la longitud I3v correspondientes *)
Quaterniones se redondean si alguno de	ClearAll[P0,P3,Q0,Q3, I3v]
almacenan en las variables correspondientes (P0, P1, Q0, Q1) para poder ser utilizados	(* Posición del elemento terminal *)
posteriormente, además, se imprimen dichos valores.	posicion = {11,0,-1};
	(* Componentes ortogonales de Rp *)
	Rp1 = Rp[[1]]; Rp2 = Rp[[2]]; Rp3 = Rp[[3]];
	(* Se determinan "P", "Q" y "l3v" utilizando los siguiente valores iniciales *)
	iP0 = .7071; iP3 = .7071; iQ0 = .866; iQ3 = .5; II3v=0;
	(* Se almacena la solución en la variable sol *)
	sol = FindRoot[{Rp1 == posicion[[1]], Rp2 == posicion[[2]], Rp3 == posicion[[3]], P0^2+P3^2 == 1, Q0^2+Q3^2 == 1}, {P0,iP0},{P3,iP3},{Q0,iQ0}, {Q3,iQ3},{I3v,iI3v},MaxIterations->50]
	(* Se asignan los valores de la solución a las variables correspondientes *)
	P0=P0/.sol; P3=P3/.sol; Q0=Q0/.sol; Q3=Q3/.sol; 3v=I3v/.sol;
	(* Se redondean los valores de los Quaterniones de la solución *)
	If[Chop[Abs[P0]-1]==0,P0=1;P3=0, f[Chop[Abs[P3]-1]==0,P0=0;P3=1]]; If[Chop[Abs[Q0]-1]==0,Q0=1;Q3=0, If[Chop[Abs[Q3]-1]==0,Q0=0;Q3=1]];
	(* Se imprimen los resultados *)
	Print["P0 = ",P0," ","P3 = ",P3," Q0 = ",Q0," ", "Q3 = ",Q3, " ","I3v = ",I3v]
De la misma forma que se realizó la modelación de la configuración no deformada, ahora se presenta el cálculo de la	MODELADO DE LA CONFIGURACION DEFORMADA
forma final de las bases locales de los	Modelo para la configuración deformada.
eslabones, las cuales se obtuvieron en el capitulo anterior y son:	ClearAll[r10,r13,r20,r21,r40,r41,L1,L2,L3,L3v,L4,P0, P3,Q0,Q3];
--	---
$e^{ii} = o(rt^*P, e)$	(* Posición no deformada conocida *)
	$P = \{P0,0,0,P3\}; Q = \{Q0,0,0,Q3\};$
$\mathbf{r}_{j} = \rho(\mathbf{r}_{1} \cdot \mathbf{r}_{2} \cdot \mathbf{P} \cdot \mathbf{Q}, \mathbf{e}_{j})$	(* Quaterniones para rotar a la configuración
$\mathbf{g}_{j}^{*} = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{e}_{j})$	deformada *)
$\mathbf{h}_{j}^{\mathrm{VI}} = \rho(\mathbf{r}1^{*}\mathbf{r}2^{*}\mathbf{r}4^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{e}_{j})$	r1 = {r10,0,0,r13}; (* Rotación Eslabón 1*) r2 = {r20,0,0,r23}; (* Rotación Eslabón 2*)
También on ante cano, se definan las normes	r3 = {r30,0,0,r33}; (* Rotación Eslabón 3*)
unitarias de los Quaterniones utilizados.	r4 = {r40,0,0,r43}; (* Rotación Eslabón 4*)
almacenándolas en las variables "normas3"	(* Normas unitarias de los Quaterniones *)
y normasa, que incluyen las normas unitarias de los Quaterniones	normas3 = Flatten[{normas1, r13->(1-r10^2)^(1/2),
correspondientes a la configuración no	$r23 - (1-r20^{2})^{(1/2)}, r43 - (1-r40^{2})^{(1/2)};$ normas4 = Flatteni{normas2 (1-r10^2)^{(1/2)->r13
deformada.	(1-r20^2)^(1/2)->r23, (1-r40^2)^(1/2)->r43}];
	(* Se obtienen los productos de Quaterniones necesarios para rotar las bases *)
	r1P = ProductoQ[r1,P];
	r1r2PQ = ProductoQ[r1,ProductoQ[r2,
	r1r2r4PQ = ProductoQ[r,Q]]];
	ProductoQ[r4, ProductoQ[P,Q]]]];
	(* Se obtiene la forma final de las bases locales de los eslabones *)
	(* Base local del eslabón 1 *)
	e12 = Expand[Expand[Rotacion[r1P,e1]]//.normas3]
	//.normas4; e22 = Expand[Expand[Rotacion[r1P,e2]]//.normas3]
	//.normas4;
	e32 = Expand[Expand[Rotacion[r1P,e3]]//.normas3] //.normas4;
	(* Base local del eslabón 2 *)
	f14 = Expand[Expand[Rotacion[r1r2PQ,f1]]
	//.normas3j//.normas4; f24 = ExpandiExpandiRotacionir1r2PQ.f211
	//.normas3]//.normas4;
	f34 = Expand[Expand[Rotacion[r1r2PQ,f3]] //.normas3]//.normas4;
	(* Base local del eslabón 3 *)
	g14 = Expand[Expand[Rotacion[r1r2PQ,g1]]
	//.normas3///.normas4; o24 = ExpandiExpandiRotacionir1r2PO.o211
	//.normas3]//.normas4;
	g34 = Expand[Expand[Rotacion[r1r2PQ,g3]] //.normas3]//.normas4;

$ \begin{array}{llllllllllllllllllllllllllllllllllll$		(* Base local del eslabón 4 *)
<pre>//.normas3//.normas4; h25= Expand[Expand[Rotacion[r1r2r4PQ,h2]] //.normas3//.normas4; h35 = Expand[Expand[Rotacion[r1r2r4PQ,h3]] //.normas3//.normas4; h35 = Expand[Rotacion[r1r2r4PQ,h3]] //.normas3//.normas4; h35 = Expand[Rotacion[Rotacion[r1r2r4PQ,h3]] //.lormas3//.normas4; h35 = Expand[Rotacion[Rotacion]r1r2r4PQ,h3]] //.lormas3//.normas4; h35 = Expand[Rotacion[Rotacion]r1r2r4PQ,h3]] //.lormas3//.normas4; h35 = Expand[Expand[Policy] b3 = L3*(-g34); (* Esiabón 3 ripio] b3 = L3*(-g34); (* Esiabón 3 ripio] b3 = L3*(-g34); (* Esiabón 3 variable) b4 = L4*(-h35); (* Pinza *) b5 = L5*h16; (* Pinza *) b5 = L5*h16; (* Pinza *) constantes = [0°P3->A1, P0*2->(1-A6)/2, r20*r23->A1, r00*2->(1-A6)/2, r20*r23->A1, r20*2->(1-A6)/2, r20*r23->A1, r20*2->(1-A6)/2, r20*r23->A1, r20*2->(1-A6)/2, r20*r23->A1, r20*2->(1-A6)/2, r20*r23->A1, r20*2->(1-A6)/2, r20*r23->A1, r20*2->(1-A6)/2, r20*r23->A1, r20*2->(1-A6)/2, r20*r23->A1, r20*2->(1-A6)/2, r20*r23->A1, r20*2->(1-A6)/2, r20*r23->A1,</pre>		h15 = Expand[Expand[Rotacion[r1r2r4PQ,h1]]
$\begin{array}{l lllllllllllllllllllllllllllllllllll$		//.normas3]//.normas4;
$ \frac{(".normas3]/!.normas4;}{h35 = Expand[Expand[Rotacion[r1r2r4PQ,h3]]} \\ \frac{("Normas3]/!.normas4;}{h35 = Expand[Expand[rotacion[rotacion[r1r2r4PQ,h3]]} \\ \frac{("Normas3]/!.normas4;}{h35 = Expand[Expand[rotacion[ro$		n25= Expand[Expand[Rotacion[r1r2r4PQ,h2]]
[135 = expand[//.normas3j//.normas4;
Una vez obtenidas las bases locales en su forma final, se obtienen las ecuaciones que representan los esiabones del manipulador y la posición del elemento terminal como: b ₁ = L ₁ *e ¹ ₁ = L ₁		// normanalexpandicoracion[r1r2r4PQ,h3]]
Una vez obtenidas las bases locales en su forma final, se obtienen las ecuaciones que representan los eslabones del manipulador y la posición del elemento terminal como: b, = L,* e ¹ = L,*e ¹ =		//.normassj//.normas4;
forma final, se obtienen las ecuaciones que representan los eslabones del manipulador y la posición del elemento terminal como: b, = L,* e ¹ ₁ = L,*e12; (Eslabón 1) b ₂ = L ₂ *f ¹ ₁ = L,*e12; (Eslabón 2) b ₃ = L ₃ *(-g ¹ ₃) = -L ₃ *g34; (Eslabón 3 fijo) b ₃ = (L ₃ +L ₃)'(-g ¹ ₃ ') = - (-(L ₃ +L ₃)'(-g ¹ ₃) = - (-(L ₃ +L ₃)'(-g ¹ ₃) = - (-(L ₃ +L ₃)'(-g ¹ ₃) = - (-(L ₃ +L ₃)'(-g ¹ ₃) = - (-(L ₃ +L ₃)'(-g ¹ ₃) = - (-(L ₃ +L ₃)'(-g ¹ ₃) = - (-(L ₃ +L ₃)'(-g ¹ ₃) = - (-(L ₃ +L ₃)'(-g ¹ ₃) = - (-(L ₃ +L ₃)'(-g ¹ ₃) = - (-(L ₃ +L ₃)'(-g ¹ ₃) = - (-(L ₃ +L ₃))'(-g ¹ ₃) = - (-(L ₃ +L ₃))'(-g ³ ₃) = - (-(L ₃ +L ₃))'(-g ³ ₃) = - (-(L ₃ +L ₃))'(-g ³ ₃) = - (-(L ₃ +L ₃))'(-g ³ ₃) = - (-(L ₃ +L ₃))'(-g ³ ₃) = - (-(L ₃ +L ₃))'(-g ³ ₃) = - (-(L ₃ +L ₃))'(-g ³ ₃) = - (-(L ₃ +L ₃))'(-g ³ ₃) = - (-(L ₃ +L ₃))'	Una vez obtenidas las bases locales en su	(* Vectores que definen los eslabones del
representan los eslabones del manipulador y la posición del elemento terminal como: b, = L,* $e_1^{H} = L,* e_12;$ (Eslabón 1) b_2 = L_2* f_1^{V} = L_2* f_14; (Eslabón 2) b_3 = L_3* (- g_3^{V}) = -L_3* g34; (Eslabón 3 ruirable) b_3 = (L_3,+L_3,)* (- g_3^{V}) = -(L_3,+L_3,)* (Eslabón 3 variable) b_4 = L_4* (- h_3^{V}) = -L_4* h35; (Pinza) b_5 = L_5* h_1^{V} = L_4* h15; (Pinza) b_5 = L_5* h_1^{V} = L_4* h15; (Pinza) constantes para simplificar *) (* Constantes = (P0*P3->A1, P0*2->(1-A2)/2, C0*Q3->A3, Q0*2->(1-A4)/2, r10*r13->A5, r10*2->(1-A4)/2, r10*r13->A5, r10*2->(1-A6)/2, r20*r23->A7, r20*2->(1-A6)/2, r20*r23->A7, r20*2->(1-A6)/2, r20*r23-P3, r40*2->(1-A6)/2, r20*r23-P3, r40*2->(1-A6)/2, r20*r23-P3, r40*2->(1-A6)/2, r20*r23-P3, r20*r23->(1-A6)/2, r20*r23-P3, r20*r23->(1-A6)/2, r20*r23-P3, r20*r23->(1-A6)/2, r20*r23-P3, r20*r23->(1-A6)/2, r20*r23-P3, r20*r23->(1-A6)/2, r20*r23-P3, r20*r23->(1-A6)/2, r20*r23-P3, r20*r23->(1-A6)/2, r20*r23-P3, r20*r23->(1-A6)/2, r20*r23-P3, r20*r23->(1-A6)/2, r20*r23*r23, r20*r23->(1-A6)/2	forma final, se obtienen las ecuaciones que	manipulador *)
la posición del elemento terminal como: b, = L,* e,!' = L,*e12; (Eslabón 1) b_ = L_* e,!' = L,*e12; (Eslabón 1) b_ = L_* f,!' = L_*f14; (Eslabón 2) b_ = L_*(-g1^V) = - L_*g34; (Eslabón 3 fijo) b_ = L_*(-g1^V) = - L_*g34; (Eslabón 3 variable) b_ = L_*(-b_Y^V) = - L_*h35; (Pinza) b_ = L_*(-b_Y^V) = - L_*h35; (Pinza) b_ = L_*h_Y^V = L_*h16; (Pinza) b_ = L_*h_Y^V = L_*h16; (Pinza) constantes = {P0*P3->A1, P0*2->(1-A2)/2, Constantes = {P0*P3->A1, P0*2->(1-A2)/2, C0*C12->A7, Z0*2->(1-A2)/2, C0*C12->C1+A2)/2, C0*C2->C1+A2)/2, C0*C2->C1+A2)/2, Constantes], Constantes = {P0*P3->A1, P0*2->(1-A2)/2, Constantes], Constantes = {P0*P3->A1, P0*2->(1-A2)/2, C0*C2->C1+A2)/2, C0*C2->C1+A2)/2, C0*C2->C1+A2)/2, Constantes], Constantes], Constantes], Constantes], Constantes], Constantes], Constantes], Constantes], Constantes], Constantes], Constantes], Constantes], Constantes], Const	representan los eslabones del manipulador v	
$b_{1} = L_{1}^{*} e_{1}^{!1} = L_{1}^{!1} e_{1}^{!1} e_{1}^{!1} = L_{1}^{!1} e_{1}^{!1} e_{1}^{!1} e_{1}^{!1} e_{1}^{!1} = L_{1}^{!1} e_{1}^{!1$	la posición del elemento terminal como:	b1 = L1*e12; (* Eslabón 1 *)
b ₁ = L ₁ * e ₁ ¹¹ = L ₁ *e ₁ ² ; (Eslabón 1) b ₂ = L ₂ * f ₁ ^V = L ₂ *f ₁₄ ; (Eslabón 2) b ₃ = L ₃ *(-g ₁ ^V) = -L ₃ *g ₃₄ ; (Eslabón 3 fijo) b ₃ v = (L ₃ v+L ₃), [*] (g ₁ ^V) = -(L ₃ v+L ₃), [*] (g ₁ ^V) = -(L ₃ v+L ₃), [*] (g ₁ ^V) = -(L ₃ v+L ₃), [*] (g ₁ ^V) = -(L ₃ v+L ₃), [*] (g ₁ ^V) = -(L ₃ v+L ₃), [*] (g ₁ ^V) = -(L ₃ v+L ₃), [*] (g ₁ ^V) = -(L ₃ v+L ₃), [*] (f ₁ ^V) = -(L ₄ *h ₃₅ ; (Pinza)) b ₄ = L ₄ *(-h ₃ ^V) = -L ₄ *h ₃₅ ; (Pinza)) b ₅ = L ₅ *h ₁ ^V = L ₆ *h ₁₅ ; (Pinza)) b ₅ = L ₅ *h ₁ ^V = L ₆ *h ₁₅ ; (Pinza)) c(Posición del elemento terminal)) r _p = b ₁ + b ₂ + b ₃ + b ₃ + b ₄) Estas expresiones son simplificadas sustituyendo por constantes a aquellos términos que se repiten dentro de las diferentes ecuaciones, como se puede observar. (* Se simplifican las expresiones *) ti = Expand[Expand[Expand[b2//.constantes] //.constantes]; b ₅ = Expand[Expand[Expand[b2//.constantes] //.constantes]; b ₅ = Expand[Expand[Expand[b2//.constantes] //.constantes]; constantes]; constantes]; constantes]; constantes]; constantes]; constantes]; b ₅ = Expand[Expand[Expand[b2//.constantes] //.constantes]; b ₅ = Expand[Expand[b2//.constantes] //.constantes]; b ₅ = Expand[Expand[b2/		b2 = L2*f14; (* Eslabón 2 *)
$b_{2} = L_{2}^{*} f_{1}^{1/2} = L_{2}^{*} h_{1}^{2} + L_{2}^{*} h_{2}^{2} + L_{2$	$b_1 = L_1^* e_1^{11} = L_1^* e_12;$ (Eslabón 1)	b3 = L3*(-g34); (* Eslabón 3 fijo *)
$b_{3} = L_{3}^{*}(-g_{3}^{*}') = -L_{3}^{*}g_{3}^{*}($ (Eslabón 3 fijo) $b_{3v} = (L_{3v}+L_{3v})^{*}(-g_{3}^{*}') = -L_{3v}^{*}h_{3}^{*}($ (Eslabón 3 variable) $b_{4} = L_{4}^{*}(-h_{3}^{*}') = -L_{4}^{*}h_{3}^{*}($ (Eslabón 3 variable) $b_{4} = L_{4}^{*}(-h_{3}^{*}') = -L_{4}^{*}h_{3}^{*}($ (Eslabón 3 variable) $b_{5} = L_{5}^{*}h_{1}^{*} = L_{3}^{*}h_{1}^{*} = L_{4}^{*}h_{3}^{*}($ (Pinza) $(^{Posición del elemento terminal)$ $r_{p} = b_{1} + b_{2} + b_{3} + b_{3} + b_{4}$ Estas expresiones son simplificadas sustituyendo por constantes a aquellos términos que se repiten dentro de las diferentes ecuaciones, como se puede observar. $(^{*} Se simplifican las expresiones ^{*})$ $b_{1} = Expand[Expand[b1//.constantes]/.constantes]$ $(^{*} Vector que define posición del elemento terminal^{*})$ $r_{p} = Factor[b1+b2+b3+b3v+b4];$ En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones ismplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones ismplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones ismplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones ismplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones ismplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones ismplificadas del manipulador, así como, las constantes utilizadas en crealizar (terminate a constantes a la que ($b_2 = L_2^* f_1^{IV} = L_2^* f_1^{I4};$ (Eslabón 2)	D3v = (L3v+13v)*(-g34); (* Eslabon 3 variable *) b4 = L4*(-h35); (* Pinze *)
b ₃ , = (L ₃ ,+L ₃ ,)'(-g ₃ ^{TV}) = -(L ₃ ,+L ₃ ,)'g34; (Esiabón 3 variable) b ₄ = L ₄ '(-h ₃ ') = -L ₄ *h35; (Pinza) b ₅ = L ₆ *h ^Y = L ₄ *h16; (Pinza) (Posición del elemento terminal) r _p = b ₁ + b ₂ + b ₃ + b ₄ + b ₄ Estas expresiones son simplificadas sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a aquellos términos que se repiten dentro de las sustituyendo por constantes a quellos términos que se repiten dentro de las sustituyendo por constantes a quellos términos que se repiten dentro de las sustituyendo por constantes a quellos términos que se repiten dentro de las (* Se simplificadas (Expand[Expand[Expand[b5//.constantes] //.constantes], b5 = Expand[Expand[Expand[b5//.constantes] //.constantes]; b5 = Expand[Expand[b5//.constantes] //.constantes]; b5 = Expand[Expand[b5//.constantes] //.constantes]; b5 = Careciones optimizadas para la posición deformada. cuearial(r10,r13,r20,r23,r40,r43,L1,L2,L3v,L4); constantes cimplificadas	$b_3 = L_3^*(-g_3^{1V}) = -L_3^*g_34;$ (Estabón 3 filo)	b5 = L5*h15; (* Pinza *)
 - (L₃, +L₃,)'g34; (Esiabón 3 variable) b₄ = L₄*(-h₃[*]) = -L₄*h35; (Pinza) b₅ = L₅*h[*] = L₅*h15; (Pinza) constantes = {P0*P3->A1, P0^2->(1-A2)/2, Q0*Q3->A3, Q0^2->(1-A4)/2, r10*r13->A5, r10*2->(1-A6)/2, r10*r13->A5, r10*2->(1-A6)/2, r20*r23->A7, r20*2->(1-A8)/2, r40*r43->A9, r40*2->(1-A10)/2, A2*A6->4*A1*A5+B1, A2*A5->B2-A1*A6, A4*A8->4*A3*A7+B3, A4*A7->B4-A3*A8, B1*B3->C1+4*B2*B4, B1*B4->C2-B2*B3); Estas expresiones son simplificadas sustituyendo por constantes a aquellos términos que se repiten dentro de las diferentes ecuaciones, como se puede observar. (* Se simplifican las expresiones *) b1 = Expand[Expand[Expand[b1//.constantes]//.constantes]; b5 = Expand[Expand[Expand[b5//.constantes]; b5 = Expand[Expand[Expand[b5//.constantes]] b2 = Expand[Expand[Expand[b5//.constantes] b3 = Expand[Expand[Expand[b5//.constantes]; b4 = Expand[Expand[Expand[b5//.constantes]] b5 = Expand[Expand[Expand[b5//.constantes]] constantes willizadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que romismara se utilizadas en dichas ecuaciones and utilizadas en dichas ecuaciones and utilizadas en dichas ecuaciones are realized to btenido en la sección es la que romismara se utilizadas en dichas ecuaciones area utilizadas en dichas ecuaciones area realized to terminal and the sección es la que romismara as explicadas en dichas ecuaciones area realized to terminal and the sección es la que romismara se a terminar la que romismara esta parte se presenta el resultado ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones area realized to terminal and terminates and terminate andificadas en dichas ecuaciones simplificadas del manipulador,	$b_{1,1} = (L_{1,1}+L_{1,1})^* (-g_1^{V}) =$	
<pre>b_a = L_a*(-b_3^V) = - L_a*h35; (Pinza) b_b = L_b*h^V; = L_b*h35; (Pinza) constantes = {P0*P3->A1, P0*2->(1-A2)/2, Q0*Q3->A3, Q0*2->(1-A4)/2, r10*r13->A5, r10*2->(1-A6)/2, r20*r23->A7, r20*2->(1-A8)/2, r40*r43->A9, r40*2->(1-A10)/2, A2*A5->4*A*A*A7+B3, A4*A7->B4-A3*A8, B1*B3->C1+4*B2*B4, B1*B4->C2-B2*B3); (* Se simplifican las expresiones *) b1 = Expand[Expand[b1//.constantes]//.constantes] b1 = Expand[Expand[b1//.constantes], b5 = Expand[Expan</pre>	- (L,+L,)*234: (Esishón 3 varishis)	(* Constantes para simplificar *)
b _a = L _a * h ₁ [*] = L _a *h15; (Pinza) b _b = L _a * h ₁ [*] = L _a *h15; (Pinza) c = b ₁ * b ₁ * = L _a *h15; (Pinza) c = b ₁ * b ₂ * b ₃ * b ₃ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ c = constantes	b. sl. *(. b ^Y) = . i *h26: / Dines	 constantes = {P0*P3->A1 P0^2->(1-A2)/2
D _b = L _b * D ₁ * = L _b *D15; (Pinza) (Pinza) (Posición del elemento terminal) r _b * = b ₁ + b ₂ + b ₃ + b ₄ r _p = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ (Posición del elemento terminal) r _p = b ₁ + b ₂ + b ₃ + b ₃ + b ₄ Estas expresiones sustituyendo por constantes a aquellos términos que se repiten dentro de las diferentes ecuaciones, como se puede observar. (* Se simplifical as expresiones *) términos que se repiten dentro de las diferentes ecuaciones, como se puede observar. (* Se simplifical as expresiones *) terminos que se repiten dentro de las (* Se simplifical as expresiones *) terminos que se repiten dentro de las (* Se simplifical as expresiones *) terminos que se repiten dentro de las (* Se simplifical as expresiones *) total = Expand[Expand[Expand[b2//.constantes] //.constantes]//.constantes] //.constantes]//.constantes]; b5 = Expand[Expand[Expand[b5//.constantes]; (* Vector que define posición del elemento terminal*) rp = Factor[b1+b2+b3+b3v+b4]; En esta parte se presenta el resultado obteni	$w_4 = w_4 (-u_3) (-w_4 u_3) (-w_2 u_3)$	Q0°Q3->A3. Q0^2->(1-A4)/2.
<pre>(Posición del elemento terminal) r_p = b₁ + b₂ + b₃ + b₄ Estas expresiones son simplificadas sustituyendo por constantes a aquellos términos que se repiten dentro de las diferentes ecuaciones, como se puede observar.</pre>	$D_{5} = L_{5} n_{1}^{*} = L_{5} n_{1}^{*}$ (Pinza)	r10*r13->A5, r10^2->(1-A6)/2.
<pre>(Posicion dei elemento terminal) r_p = b₁ + b₂ + b₃ + b₃ + b₄ Estas expresiones son simplificadas sustituyendo por constantes a aquellos términos que se repiten dentro de las diferentes ecuaciones, como se puede observar.</pre>		r20*r23->A7, r20^2->(1-A8)/2.
rp = b1 + b2 + b3 + b3, + b4A2*A6->4*A1*A5+B1, A2*A5->B2-A1*A6, A4*A8->4*A3*A7+B3, A4*A7->B4-A3*A8, B1*B3->C1+4*B2*B4, B1*B4->C2-B2*B3);Estas expresiones son simplificadas sustituyendo por constantes a aquellos términos que se repiten dentro de las diferentes ecuaciones, como se puede observar.(* Se simplifican las expresiones *)b1 = Expand[Expand[b1//.constantes]//.constantes]; b2 = Expand[Expand[b2//.constantes]; b5 = Expand[Expand[b5//.constantes]; b5 = Expand[Expand[b5//.constantes]; i55 = Expand[Expand[b5//.constantes]; i55 = Expand[Expand[b5//.constantes]; i56 = Expand[Expand[b5//.constantes]; i57 = Expand[Expand[b1+b2+b3+b3v+b4];En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones. Esta sección es la que ronjamente, ce utilizadas en dichas ecuaciones. Esta sección es la que ronjamente, ce utiliza para realizarda iconstantes a quierEcuaciones Simplificadas *)	(Posición del elemento terminal)	r40*r43->A9, r40^2->(1-A10)/2,
rp = D1 + D2 + D3 + D3 + D4 A4*A8->4*A3*A7+B3, A4*A7->B4-A3*A8, B1*B3->C1+4*B2*B4, B1*B4->C2-B2*B3); Estas expresiones son simplificadas sustituyendo por constantes a quellos términos que se repiten dentro de las diferentes ecuaciones, como se puede b1 = Expand[Expand[b1//.constantes]//.constantes]; b2 = Expand[Expand[b2//.constantes]; b5 = Expand[Expand[b2//.constantes]; b5 = Expand[Expand[b5//.constantes]; b5 = Expand[Expand[b5//.constantes]; b5 = Expand[Expand[b1+b2+b3+b3v+b4]; rp = Factor[b1+b2+b3+b3v+b4]; En esta parte se presenta el resultado constantes optimizadas para la posición obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas clearAll[r10,r13,r20,r23,r40,r43,L1,L2,L3v,L4]; (* Ecuaciones Cimplificades *) (* Ecuaciones Cimplificades *)		A2*A6->4*A1*A5+B1, A2*À5->B2-A1*A6,
Estas expresiones son simplificadas sustituyendo por constantes a aquellos términos que se repiten dentro de las diferentes ecuaciones, como se puede observar.	$r_p = D_1 + D_2 + D_3 + D_{3v} + D_4$	A4*A8->4*A3*A7+B3, A4*A7->B4-A3*A8,
<pre>sustituyendo por constantes a aquellos términos que se repiten dentro de las diferentes ecuaciones, como se puede observar.</pre> (* Se simplifican las expresiones *) b1 = Expand[Expand[b1//.constantes]//.constantes]; b2 = Expand[Expand[b2//.constantes]; b5 = Expand[Expand[b5//.constantes]; b5 = Expand[Expand[b5//.constantes]; b5 = Expand[Expand[b5//.constantes]; //.constantes]//.constantes]; b5 = Expand[Expand[b5//.constantes]; b5 = Expand[Expand[b5//.constantes]; constantes utilizadas en cichas ecuaciones Esta sección es la que expanse e a utilizadas en cichas ecuaciones. Esta sección es la que (* Ecuaciones Emplificadas *)		B1*B3->C1+4*B2*B4, B1*B4->C2-B2*B3};
términos que se repiten dentro de las diferentes ecuaciones, como se puede observar. (* Se simplifican las expresiones *) b1 = Expand[Expand[b1//.constantes]//.constantes]; b2 = Expand[Expand[b2//.constantes]; b5 = Expand[Expand[b5//.constantes]; b5 = Expand[expand[b5//.constantes]; constantes utilizadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que expensiones e a utiliza para replizar la (* Ecuaciones Simplificadas *)	Loldo expresiones son simplificadas	
diferentes ecuaciones, como se puede b1 = Expand[Expand[b1//.constantes]//.constantes]; b2 = Expand[Expand[Expand[b2//.constantes]; b5 = Expand[Expand[b5//.constantes]; b5 = Expand[Expand[Expand[b5//.constantes]; b5 = Expand[Expand[b5//.constantes]; b5 = Expand[Expand[b5//.constantes]; b6 = Expand[Expand[b5//.constantes]; b7 = Factor[b1+b2+b3+b3v+b4]; rp = Factor[b1+b2+b3+b3v+b4]; En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que replicar parte resultado (1) ClearAll[r10,r13,r20,r23,r40,r43,L1,L2,L3v,L4]; ClearAll[r10,r13,r20,r23,r40,r43,L1,L2,L3v,L4]; ClearAll[r10,r13,r20,r23,r40,r43,L1,L2,L3v,L4];	términos que se repiten dentro de las	(* Se simplifican las expresiones *)
observar. b2 = Expand[Expand[Expand[b2//.constantes], //.constantes], //.constantes], //.constantes]; b2 = Expand[Expand[Expand[b2//.constantes], //.constantes], //.constantes], //.constantes]; b5 = Expand[Expand[Expand[b5//.constantes], //.constantes], //.constantes]; c* Vector que define posición del elemento terminal*) rp = Factor[b1+b2+b3+b3v+b4]; En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que replicar la (* Ecuaciones Simplificadas *) ClearAll[r10,r13,r20,r23,r40,r43,L1,L2,L3v,L4];	diferentes ecuaciones, como se puede	b1 = Expand/Expand/b1//.constantes//.constantes)
//.constantes]//.constantes]; b5 = Expand[Expand[Expand[b5//.constantes] //.constantes]//.constantes]; (* Vector que define posición del elemento terminal*) rp = Factor[b1+b2+b3+b3v+b4]; En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que repliamente se utiliza para realizar la (* Ecuaciones Simplificadas *)	observar	b2 = Expand[Expand[Expand[b2//.constantes]
b5 = Expand[Expand[Expand[b5//.constantes] //.constantes]//.constantes]; (* Vector que define posición del elemento terminal*) rp = Factor[b1+b2+b3+b3v+b4]; En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que replimente es utilizar para replizar la (* Ecuaciones Simplificadas *)		//.constantes]//.constantes];
<pre>//.constantes]//.constantes]; //.constantes]; (* Vector que define posición del elemento terminal*) rp = Factor[b1+b2+b3+b3v+b4]; En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que reprisente es utiliza parte replizar la (* Ecuaciones Simplificadas *)</pre>		b5 = Expand[Expand[Expand[b5//.constantes]
(* Vector que define posición del elemento terminal*) rp = Factor[b1+b2+b3+b3v+b4]; En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que repliamente es utilizar para realizar la (* Ecuaciones Simplificadas *)		//.constantes]//.constantes];
rp = Factor[b1+b2+b3+b3v+b4]; En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que reprisente persistent del content del		(* Vector que define posición del elemento terminal*)
rp = Factor[b1+b2+b3+b3v+b4]; En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que propiamente es utiliza para realizar la (* Ecuaciones Simplificadas *)		
En esta parte se presenta el resultado obtenido en la sección anterior, es decir, las ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que replamente se utiliza para replizar la (* Esusciones Simplificadas *)		rp = Factor[b1+b2+b3+b3v+b4];
obtenido en la sección anterior, es decir, las deformada. ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que replamente es utilizar para replizar la (* Ecuaciones Simplificadas *)	En esta parte se presenta el resultado	Ecuaciones optimizadas para la posición
ecuaciones simplificadas del manipulador, así como, las constantes utilizadas en dichas ecuaciones. Esta sección es la que propiamente es utilizar para realizar la (* Ecuaciones Simplificadas *)	obtenido en la sección anterior, es decir, las	deformada.
así como, las constantes utilizadas en dichas ClearAll[r10,r13,r20,r23,r40,r43,L1,L2,L3v,L4]; ecuaciones. Esta sección es la que propiamente se utiliza para realizar la (* Ecuaciones Simplificadas *)	ecuaciones simplificadas del manipulador,	
ecuaciones. Esta sección es la que propiamente se utiliza para realizar la (* Ecuaciones Simplificados *)	así como, las constantes utilizadas en dichas	ClearAll[r10,r13,r20,r23,r40,r43,L1,L2,L3v,L4];
nronigmente se utiliza nara realizar la (* Ecuaciones Simplificadas *)	ecuaciones. Esta sección es la que	
propriemente se uniza para realizar la (Ecuaciones Simplinicadas)	proplamente se utiliza para realizar la	(* Ecuaciones Simplificadas *)
simulación de la cinemática directa del	simulacion de la cinemática directa del	
manipulador de tres grados de libertad, ya bi = (B1*L1, -2*B2*L1, U); (* Esiabon 1 *)	manipulador de tres grados de libertad, ya	D1 = (B1*L1, -2*B2*L1, V); (* Esiabon 1 *) b2 = /C1*L2, -2*C2*L2, 0\+ (* Esiabén 2 *)
que es la que contiene las ecuaciones ya $DZ = \{U1^{-1}LZ, \cdot Z^{-}UZ^{-1}LZ, U\}; (* Estadon Z^{-})$ elimplificadas. Como al realizar la elimplación h2 = $(0, 0, -1, 2)$; (* Estadon Z^{-})	que es la que contierre las ecuaciones ya	$ UZ = \{U \mid LZ, *Z \mid UZ \mid LZ, 0\}, (* Esiabon Z^*)$ $ UZ = \{0, 0, -1, 2\}, (* Esiabón 2, 5in *)$
es necesario evaluar numéricamente las h $3v = \{0, 0, -L3\},$ (Estaboli 5 lijo) es necesario evaluar numéricamente las h $3v = \{0, 0, -L3v+13v\}$ (* Felabón 3 variable *)	es necesario evaluar numéricamente las	b3v = {0, 0, -L3v, -L3v, } (Estaboli 5 lij0) b3v = {0, 0, -{L3v+l3v}} (* Felabón 3 variable *)
ecuaciones del manipulador, se genera un	annagenes del masiculador se server un	

respaldo de dichas ecuaciones en esta misma sección, almacenándolas en nuevas variables.	<pre>(* Pinza *) b4 = {0, 0, -L4}; b5 = {(4*A9*C2 - A10*C1)*L5, 2*(A9*C1 + A10*C2)*L5, 0}; (* Posición del elemento terminal *) rp = {B1*L1 + C1*L2, -2*(B2*L1 + C2*L2), -(L3+I3v+L3v+L4)}; (* Constantes de las ecuaciones *) A1 = P0*P3; A2 = 1-2*P0^2; A3 = Q0*Q3; A4 = 1-2*Q0^2; A5 = r10*r13; A6 = 1-2*r10^2; A7 = r20*r23; A8 = 1-2*r20^2; A9 = r40*r43; A10 = 1-2*r40^2; B1 = A2*A6 - 4*A1*A5; B2 = A2*A5 + A1*A6; B3 = A4*A8 - 4*A3*A7; B4 = A4*A7 + A3*A8; C1 = B1*B3 - 4*B2*B4; C2 = B1*B4 + B2*B3; (* Longitud de los eslabones del manipulador *) base = 7; L1 = 6; L2 = 5; L3 = 1; L4 = 1; L5 =2; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; z3v=b3v; z4=b4; z5=b5; v==***********************************</pre>
Finalmente, en esta parte del programa, se	CINEMATICA DIRECTA
nos permitirán observar la simulación de la cinamática directa del manipulador en	Graficación para visualizar la simulación.
cinernatica directa del manipulador en cuestión.	ClearAll[r10,r13,r20,r23,r40,r43,L3v];
Para lograr generar las gráficas adecuadas, para poder observar la simulación, primero generamos mediante la función SurfaceGrafica una gráfica que contenga el	(* Se define el espacio tridimensional en que se graficará el manipulador, que se usará como gráfica base *)
espacio donde se mostrará el manipulador con una superficie simulando el piso y con la indicación de los ejes coordenados, la cual	(* Suma de la longitudes de eslabones *) L235 = L2+L3+L5;
se almacena inicialmente en las variables g1 y g2.	g1=g2=SurfaceGraphics[{{0,0,0},{0,0,0},{0,0,0}}, MeshRange->{{-(L235+5),(L235+5)}, {-(L235+5),(L235+5)}},
Definimos los parametros de rotación de cada eslabón, en función de los ángulos que utilizaremos para girar cada uno de los eslaboras que serán:	AxesLabel->{Eje X,Eje Y,Eje Z}, DefaultFont->{"Negrita",12}, PlotRange->{{-(L235+5),(L235+5)},
	((1 025+5) (1 025+5)) (5 (bears 4)))

	Eslabón 1 → theta1	(* Se inicializan en cero los ángulos de rotación *)
	Eslabon 2 -> theta2	theta1 = 0' theta2 = 0' theta3 = 0' I $3y = 0$
		1 - 0, 1 - 0,
	Se generan dos ciclos For, el primero (variable de control i) servirá para controlar	(* Se definen los valores de los parámetros de las rotaciones *)
	cual estadon debera moverse, o sea, cuando	r10=Cositheta1/2 Decreel
	la función Switch v el segundo ciclo	r13=Sin[theta1/2 Degree];
	(variable de control I) nos permitirá generar	r20=Cos[theta2/2 Degree];
	varias gráficas del movimiento de cada	r23=Sin[theta2/2 Degree];
	eslabón. La variable I toma valores de 1 a 3.	r40=Cos[theta3/2 Degree];
	ya que son tres eslabones y la variable j	r43=Sin[theta3/2 Degree];
	toma valores de 1 a n, donde n es el número	
	de gráficas a generar por cada eslabón.	(* Se inicializa un ciclo para mover todos eslabones, uno a la vez *)
	tienen un valor numérico, se evaluan numéricamente las ecuaciones del	For[i=1, i<=4, i++,
	manipulador, haciendo uso del respaldo de ecuaciones realizado anteriormente	(* Se inicializa otro ciclo para generar varias gráficas para cada eslabón en diferentes posiciones*)
	Con las ecuaciones, ya numéricas, se modifica la gráfica almacenada en g2	For[j=1, j<=5, j++,
	agregándole un punto en la posición que corresponde a rp.	(* Se selecciona cual eslabón se moverá según el valor de la variable i, lo cual se logra cambiando sus parámetros de rotación *)
	De la misma forma, con las ecuaciones	
	numéricas, se determinan los diferentes	Switch[i,1, (* Se mueve eslabón 1 *)
I	puntos de la gráfica a generar, es decir, se	r10=Cos(theta1/2 Degree); r12=Sis(theta1/2 Degree);
I	genera un punto para cada inicio y final de	2 theta2-=15: (* Se mueve eslabón 2 *)
	para mostrar el vector e11 de la base del	r20=Cositheta2/2 Degree]:
	primer eslabón.	r23=Sin[theta2/2 Degree];
	P	3, L3v+=.8, (* Se mueve eslabón 3 *)
	Con los puntos anteriores, se genera la	4, theta3+=25; (* Se mueve eslabón 4 *)
	gráfica correspondiente a la posición del	r40=Cos[theta3/2 Degree];
	manipulador, utilizando para ello la función	r43=Sin[theta3/2 Degree]];
ĺ	Graphica3D, que nos permite trazar líneas	/* Se esignen valores a las ecuaciones del
	de diferente color y espesor entre los puntos	manipulador, según los valores de los parámetros.
		utilizando el respaldo de ecuaciones *)
	Finalmente se muestran en la pantalla, de	
	manera combinada, las gráficas g1, g2 y g3	b1=N[z1]; b2=N[z2]; b3=N[z3]; b3v=N[z3v];
	que se generan en cada pasada de los	D4=N[Z4]; D5=N[Z5]; IP=N[X];
	cicios, utilizando para esto la función Snow la cual envía a la pantalla objetos gráficos y	(* Se definen los puntos en el espacio tridimensional
	permite que se combinen varias gráficas.	para graficar el manipulador *)
	Se muestran, a manera de ejemplo, algunas	$punto 1 = \{0, 0, 0\},$ $punto 2 = \{0, 0, base \}$
	grancas de un proceso.	punto3 = punto2 + b1:
		punto4 = punto3 + b2;
		punto5 = punto4 + b3;

punto6 = punto5 + b3v;
punto7 = punto6 + b5;
punto8 = punto6 - b5;
punto9 = punto7 + b4;
punto10 = punto8 + b4;
punto11 = punto2 + rp;
(* Se agrega a la gráfica g2 la posición del elemento
terminal como un punto *)
. ,
g2={Graphics3D[{PointSize[0.01],
Point[punto11], RGBColor[0,1,0]]] g2];
(* Se genera la gráfica del manipulador en la
posición correspondiente *)
q3=Graphics3DI{
0
(* Base *)
AbsoluteThicknessi301.RGBColori0.1.01.
Linel(punto1.punto2)]).
(* Esiabón 1 *)
(AbsoluteThicknessi25).RGBColori1.4.0).
Linel{punto2.punto3)]}.
(* Eslabón 2 *)
{AbsoluteThickness[20].RGBColor[0.,6,1].
Linel(punto3.punto4)]}
(* Eslabón 3 fijo *)
{AbsoluteThickness[15].RGBColor[1.4.0]
Linel(punto4 punto5)]}
(* Eslabón 3 variable*)
{AbsoluteThickness[8] RGBColor(0.0.1]
Linel(punto5 punto6)]}
(* Pinza *)
{AbsoluteThickness[3].RGBColor[0.0.0]
Linef{punto7.punto8}]}.
AbsoluteThickness[3], RGBColor(0.0.0).
Linel{punto7,punto9}]}.
{AbsoluteThickness[3],RGBColor[0.0.0].
Line (punto8, punto10))})):
(* Se envían a pantalla las gráficas g1,g2 y g3
combinadas *)
·
Show[g1,g3,g2,Axes->True];
1
];



SIMULACION COMPUTACIONAL DE LA CINEMATICA



4.3.2 Cinemática inversa.

La cinemática inversa del robot SCARA, consiste en que, dada la posición y orientación del elemento terminal, se deben determinar los ángulos que deberán desplazarse cada una de las juntas del manipulador para alcanzar la posición deseada. En este caso, consiste en que dada la configuración no deformada (P_0 , P_1 , Q_0 , Q_1 , I_{3v}) y la posición del elemento terminal ($r_p = \{r_px, r_py, r_pz\}$), se deben determinar los parámetros r4₀ y r4₃ que orienten el elemento terminal y los parámetros r1₀, r1₃, r2₀, r2₃ que satisfagan la ecuación de la posición del elemento terminal, que es la siguiente:

$$\mathbf{r}_{p} = \{\mathbf{B1^{*}L_{1}} + \mathbf{C1^{*}L_{2}}, -\mathbf{2^{*}(B2^{*}L_{1}} + \mathbf{C2^{*}L_{2}}), - (\mathbf{L_{3}} + \mathbf{L_{3v}} + \mathbf{I_{3v}} + \mathbf{L_{4}})\}$$
(4.31)

Para resolver la cinemática inversa del robot SCARA, debemos plantear un sistema de 5 ecuaciones que nos permitan determinar los 4 parámetros de rotación y la traslación del eslabón 3 (L_{3v}) que permitan ubicar el elemento terminal en la posición deseada. De la ecuación de r_p obtenemos 3 ecuaciones y con las normas unitarias de los Quaterniones se completa el sistema, el cual se muestra a continuación.

 $r_{p}x = B1^{*}L_{1} + C1^{*}L_{2}$ $r_{p}y = -2^{*}(B2^{*}L_{1} + C2^{*}L_{2})$ $r_{p}z = -(L_{3} + L_{3v} + I_{3v} + L_{4}) \qquad (4.32)$ $r1_{0}^{2} + r1_{3}^{2} = 1$ $r2_{0}^{2} + r2_{3}^{2} = 1$ $r3_{0}^{2} + r3_{1}^{2} = 1$

Siendo:

A1 = P.*P.; A2 = 1-2"P." A4 = 1-2"Q." A3 = Q. Q.; A5 = r1,*r1; A6 = 1-2"r1." A7 = r2, "r2; A8 = 1-2"12.": (4.33) A9 = r4, "r4; A10 = 1-2"r4,"; B1 = A2*A8 - 4*A1*A7; B2 = A2'A7 + A1'A8: B3 = A10*A8 - 4*A7*A9: 84 = A10*A7 + A8*A9: B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*85 + B3*86; C2 = B3*B5 - 4*B4*B6;

Lo anterior nos permite ubicar el elemento terminal en el punto {r_px, r_py, r_pz} dentro de su espacio de trabajo, ahora si deseamos que el elemento terminal siga una trayectoria determinada, una de las cosas que podemos hacer es discretizar dicha trayectoria en varios puntos a lo largo de ella y determinar los parámetros de rotación de los eslabones del manipulador para cada punto, resolviendo el sistema de ecuaciones (4.32), lo que nos permitirá ubicar el elemento terminal sobre dicha secuencia de puntos. Lo anterior no permite ubicar el elemento terminal en cualquier punto de la trayectoria, solo en los puntos en que dicha trayectoria se discretizó.

Para determinar la orientación del elemento terminal (r4₀ y r4₃), debemos considerar el hecho de que solamente cambia de orientación el eje transversal de la pinza (b₅), el cual siempre es paralelo al piano x-y. Basándonos en lo anterior, podemos orientar la pinza solamente con respecto al piano x-y, lo cual podemos hacer determinando el ángulo entre b₅ y la proyección sobre el piano x-y, de la línea de referencia con la que se desea orientar la pinza. A continuación se presenta el programa desarrollado en Mathematica[®] para la cinemática inversa del manipulador tipo SCARA de cuatro grados de libertad, el cual es continuación del anterior donde se presentó la cinemática directa.

Este programa es parte integral del anterior, es decir, para que funcione deberá tenerse la parte anterior donde se presenta modelación y la cinemática directa, ya que se requieren las	CINEMATICA INVERSA
ecuaciones del manipulador. En esta sección se presenta el cálculo para tres diferentes trayectorias y desde luego la graficación para observar la simulación.	Solución de ecuaciones para diferentes trayectorias

En esta parte se determinan los parámetros de	LINEA RECTA
de puntos que pertenecen a una línea recta la	(* Se inicializan variables *)
cual se define por sus dos puntos extremos	ClearAll/r10, r13, r20, r23, r40, r43, "soir1", "soir2"
(S0, S1). Para lo anterior, primeramente	"solr4", "solL3v", "solucion", radio];
recuperamos las ecuaciones del manipulador,	
las cuales se respaldaron en otras variables,	(* Se recuperan las ecuaciones del manipulador *)
(deberá resolverse la cinemática inversa para	b1=z1; b2=z2; b3=z3; b3v=z3v; b4=z4; b5=z5;
determinar la configuración no deformada y	rp=rx; rp1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]];
ejecutar la celda donde estan las ecuaciones	(* Veriebles de centrel *)
optimizadas para la configuración deformada).	(* Variables de control *)
Para determinar la puntos sobre la travectoria	travectoria seleccionada *)
utilizamos la forma paramétrica de una recta	i=1; (* Para guardar la solución de cada punto de
que pasa por dos puntos $p_1 = \{x_1, y_1, z_1\}$ y	la trayectoria *)
$p_2 = \{x_1, y_1, z_1\}$	
	(* Valores iniciales para resolver el sistema de
$\mathbf{p}\mathbf{x} = \mathbf{x}_1 + (\mathbf{x}_2 - \mathbf{x}_1)\mathbf{t}$	ecuaciones)
$py = y_1 + (y_2 - y_1)^{t}$	ir10 = 5; $ir13 = 86$; $ir20 = 86$; $ir23 = 5$;
Con lo anterior se realiza un ciclo utilizando	(* Trayectoria en línea recta dada por los puntos
como variable de control del mismo, a la	extremos *)
variable t, con lo que determinamos puntos	
sobre la trayectoria a seguir y resolvemos el	$50 = \{4, 0, 1\}; 51 = \{0, -4, 4\};$
	(* Se inicia un ciclo para resolver el problema
$r_p x = B1^*L_1 + C1^*L_2$	cinemático inverso en los puntos en que será
r _p y= - 2*(B2*L ₁ + C2*L ₂)	discretizada la trayectoria *)
r_z = - (L_1 + L_1 +]_1 + L_2)	For[t=0, t<=1, t+=.1,
r1 ₀ - + r1 ₃ - = 1	(* Se determinan las coordenadas del punto
$r2_0^2 + r2_3^2 = 1$	correspondiente de la trayectoria ")
$r3_{3}^{2} + r3_{3}^{2} = 1$	px = So[[2]] + (S1[[2]] - So[[2]]) + (S1[[
	$p_z = S0[(3)]+(S1[(3)]-S0[(3)])^*(;$
Este sistema de ecuaciones se resuelve	
utilizando la función Findroot usando los	(* Se determina el desplazamiento del eslabón 3 *)
valores iniciales mostrados. Una vez que se	noll 2:/(i) = here (1 2+1 4) no:
resuelve el sistema de ecuaciones anterior, se	soilsvij - base-(L3+L4)-pz,
verifica si la precision de la solucion es la	(* Solución del problema cinemático inverso para
iniciales nara resolver la cinemática inversa	el punto seleccionado *)
del siguiente punto, a la solución obtenida y se	
continua el ciclo, si la solución está fuera de la	solucion[i] = FindRoot[{rp1==px, rp2==py,
precisión se envía un mensaje y se restauran	[7072+[7372==7, [2072+[2372==7], /r10 ir10\ /r13 ir12\ /r20 ir20\ /r23 ir23\
los valores iniciales continuando el ciclo hasta	MaxIterations->501:
terminar. Lina vez determinados los parámetros para	ingine and is only
ubicar el elemento terminal sobre la	(* Se verifica la precisión de la solución *)
trayectoria, se determina la rotación necesaria	
para su orientación en cada punto de la	e = 100-5; (* precision requerida *)

trayectoria (r4 ₀ , r4 ₃). En este caso deseamos que el eje transversal de la pinza (b ₆) sea perpendicular a la travectoria propuesta. Para	lf[N[Abs[px-(rp1/.solucion[i])] <e &&<br="">Abs[py-(rp2/.solucion[i])]<e],< th=""></e],<></e>
determinar los parámetros r4 ₀ y r4 ₃ , calculamos el ángulo formado entre b ₅ y la proyección de la trayectoria sobre el plano x-y, siendo el ángulo que deberá girar la pinza el complementario al ángulo calculado. Para determinar el sentido de giro de la pinza se realiza el producto cruz(producto cruz b ₅ y la proyección de la	(* Si la solución está en la precisión fijada, se definen como valores iniciales para resolver la cinemática inversa del siguiente punto de la trayectoria a los valores de la solución obtenida, si no es así se restauran los valores iniciales utilizados *)
trayectoria sobre el plano x-y.	ir10=r10/.solucion[i];ir13=r13/.solucion[i]; ir20=r20/.solucion[i];ir23=r23/.solucion[i];,
	(* Se imprime un mensaje si la solución no está en la precisión definida *)
	Print["No se encontró solución para el punto: ", {px,py,pz}];
	(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
	ir10 =.5; ir13 =.86; ir20 =.5; ir23 =.86;]; i++];
	(* Cálculo de la orientación del elemento terminal (b5 perpendicular a la trayectoria) *)
	(* Se almacenan en variables separadas la solución para cada par de parámetros *)
	For[j=1, j <i, j++,="" solr1[j]="{r10,r13}/.solucion[j];<br">solr2[j]={r20,r23}/.solucion[j]];</i,>
	(* Se determina un vector unitario en dirección de la proyección de trayectoria (tray) sobre el plano x-y *)
	tray = (S0 - S1); tray = (tray)/(Sqrt[tray[[1]]^2 + tray[[2]]^2 + tray[[3]]^2]); tray = {tray[[1]],tray[[2]],0};
	(* Se inicia un ciclo para determinar la orientación de la pinza en cada punto de la trayectoria *)
	For[j=1, j <i, j++,<="" td=""></i,>
	(* Se asigna el valor de los parámetros para el punto de la trayectoria correspondiente almacenados en las variables solr?[j] *)
	r10=solr1[j][[1]];r13=solr1[j][[2]]; r20=solr2[j][[1]];r23=solr2[j][[2]];

	(* Se ubica la pinza en la posición original *) r40=1;r43=0;
	(* Se calcula el ángulo entre el vector b5 (pinza) y la proyección de la trayectoria sobre el plano x-y *)
	cosenoang = (tray.b5)/L5;
	lf[N[cosenoang]!=0,
	ang=ArcCos[cosenoang]/Degree, ang = 90] ;
	(* El ángulo que debe girar la pinza es el complementario de "ang" y el sentido del giro lo obtenemos con un producto cruz entre la trayectoria y b5 *)
	signo = N[Cruz[tray,b5][[3]]]; signo = signo/Abs[signo];
	ang = signo*(90-ang);
	(* Se almacena en una variable los parámetros correspondientes *)
	solr4[j]=N[{Cos[ang/2*Degree],Sin[ang/2*Degree]}];]
En esta sección, al igual que en la anterior se	CIRCUNFERENCIAS O ELIPSES
resuelve la cinemática inversa para una	
secuencia de puntos sobre una trayectoria,	(* Se inicializan variables *) ClearAllfr10, r13, r20, r23, r40, r43, "soir1", "soir2"
circunferencia. El proceso que se sigue es el	"solr4", "solL3v", "solucion", radio];
cambia la manera de determinar los puntos	(* Se recuperan las ecuaciones del manipulador *)
sobre la trayectoria, que en este caso se	b1=z1; b2=z2; b3=z3; b3v=z3v; b4=z4; b5=z5;
realiza utilizando la forma paramétrica de una	rp=rx; rp1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]];
con radios radio1 y radio2.	(* Variables de control *)
	t=0; (* Para discretizar en varios puntos la
DY E radio1*Coelt1 + h	trayectoria seleccionada ")
py = radio2"Sin[t] + k;	la trayectoria *)
pr - 1,	(* Variables de control para definir la trayectoria *)
	radio1 = 8; (* Radio uno de la elipse *)
	h=0;k=-7;l=0; (* Centro de la elipse *)
	(* Valores iniciales para resolver el sistema de
	ecuaciones *)
	ir10 = .5; ir13 = .86; ir20 = .5; ir23 = .86;
	ir10 = .5; ir13 = .86; ir20 = .5; ir23 = .86;



(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
ir10 =.5; ir13 =.86; ir20 =.5; ir23 =.86;];
(* Cálculo de la orientación del elemento terminal (b5 paralelo al radio de la elipse *)
(* Se almacenan en variables separadas la solución para cada par de parámetros *)
For[j=1, j <i, j++,="" solr1[j]="{r10,r13}/.solucion[j];<br">solr2[j]={r20,r23}/.solucion[j]];</i,>
(* Se inicia un ciclo para determinar la orientación de la pinza en cada punto de la trayectoria *)
For[j=1, j <i, j++,<="" th=""></i,>
(* Se asigna el valor de los parámetros para el punto de la trayectoria correspondiente almacenados en las variables solr?[j] *)
r10=solr1[j][[1]];r13=solr1[j][[2]]; r20=solr2[j][[1]];r23=solr2[j][[2]];
(* Se ubica la pinza en la posición original *)
r40=1;r43=0;
(° Se calcula el ángulo entre el vector b5 (pinza) y la línea recta (radio) del centro de la elipse al punto de la trayectoria seleccionado *)
cosenoang = (radio[j].b5)/(Sqrt[radio[j][[1]]^2 + radio[j][[2]]^2]*L5);
lf[N[cosenoang]!=0, ang = ArcCos[cosenoang]/Degree, ang = 90] ;
(* El ángulo que debe girar la pinza es el ángulo calculado, el sentido se determina mediante el producto cruz entre b5 y el radio *)
signo = N[Cruz[b5,radio[j]][[3]]]; signo = signo/Abs[signo]; ang = signo*(ang);
(* Se almacena en una variable los parámetros correspondientes *)
soir4[j]=N{{Cos[ang/2*Degree],Sin[ang/2*Degree]}]; }



If[N[Abs[px-(rp1/.solucion[i])] <e &&<br="">Abs[py-(rp2/.solucion[i])]<e],< td=""></e],<></e>
(* Si la solución está en la precisión fijada, se definen como valores iniciales para resolver la cinemática inversa del siguiente punto de la trayectoria a los valores de la solución obtenida, si no es así se restauran los valores iniciales utilizados *)
ir10=r10/.solucion[i];ir13=r13/.solucion[i]; ir20=r20/.solucion[i];ir23=r23/.solucion[i];,
(* Se imprime un mensaje si la solución no está en la precisión definida *)
Print("No se encontró solución para el punto: ", {px,py,pz}];
(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
ir10 =.5; ir13 =.86; ir20 =.5; ir23 =.86;]; i++];
(* Cálculo del la orientación del elemento terminal, deberá conservar su orientación original *)
(* Se almacenan en variables separadas la solución para cada par de parámetros *)
For[j=1, j <i, j++,solr1[j]="{r10,r13}/.solucion[j];<br">solr2[j]={r20,r23}/.solucion[j]];</i,>
(* Se inicia un ciclo para determinar la orientación de la pinza en cada punto de la trayectoria *)
For[j=1, j <i, j++,<="" td=""></i,>
(* Se asigna el valor de los parámetros para el punto de la trayectoria correspondiente almacenados en las variables solr?[j] *)
r10=solr1[j][[1]]:r13=solr1[j][[2]]; r20=solr2[j][[1]]:r23=solr2[j][[2]];
(* Se ubica la pinza sin rotación *)
r40 = 1; r43=0;
(* Se determina el ángulo que debe rotar la pinza, que es el ángulo entre b5 y e1 *)
ang = ArcCos[(b5.{1,0,0}/L5)];

	(* Se determina el sentido de rotación de la pinza*)
	signo = Cruz[b5,{1,0,0}][[3]];
	signo = signo/Abs[signo];
	ang = signo*ang;
	(* Se almacene en une veriable el ángulo
	(Se almacena en una variable el angulo
	solr4[j] = {Cos[ang/2],Sin[ang/2]};
Lina vez determinados los parámetros de	Graficación para visualizar la simulación de la
rotación para la travectoria seleccionada,	trayectoria seleccionada.
ahora se generarán las gráficas que permitirán	
ver la simulación de la cinemática inversa del	ClearAll[r10,r13,r20,r23,r40,r43];
manipulador tipo SCARA de cuatro grados de	
libertad Al igual que en la cinemática inversa,	(* Se recuperan los parametros de rotación
se genera una gratica que servira como base	almacenados en la variable solucion(j),
almacena en las dos variables o1 v o2	inverso
posteriormente se inicia un ciclo, que se	
repetirá tantas veces como valores de los	For[j=1, j <i, j++,="" soir1[j]="{r10,r13}/.solucion[j];</td"></i,>
parámetros se hayan calculado para la	solr2[j]={r20,r23}/.solucion[j]];*)
trayectoria seleccionada (variable "i"). En cada	
pasada del ciclo se asignan los valores	(* Se define el espacio tridimensional en que se
se genera la gráfica del manipulador de la	grancara el manipulador, que se usara como
misma forma que se hizo en la cinemática	(* Suma de la longitudes de eslabones *)
directa. Se mostrarán, a manera de ejemplo,	L235 = L2+L3+L5;
algunas gráficas de una trayectoria.	
	g1=g2=SurfaceGraphics[{{0,0,0},{0,0,0},{0,0,0}},
	MeshRange->{{-(L235+5),(L235+5)}},
	{-(L230+0),(L230+0)}}, Aveal abal_>/Fia X Fia V Fia 7}
	DefaultFont->{"Negrita", 12}
	PlotRange->{{-(L235+5),(L235+5)}},
	{-(L235+5),(L235+5)},
	{-5,(base+1)}},
	BoxRatios->{1,1,1},Axes->True];
	(* Se inicia un ciclo para generar las gráficas de cada punto de la trayectoria *)
	For[j=1, j <i, j++,<="" td=""></i,>
	(* Se asigna el valor de los parámetros para el punto de la trayectoria correspondiente *)
	r10=soir1[j][[1]]; r13=soir1[j][[2]];
	r20=solr2[j][[1]]; r23=solr2[j][[2]]; r40=solr4[j][[1]]; r43=solr4[j][[2]]; L3v=solL3v[j];

/t De seiseen volenes a les sevenienes del
(* Se asignan valores a las ecuaciones del
manipulador, según los valores de los
parametros, utilizando el respaldo de
Acuaciones *)
CONTRACTOR CONTRACTOR STRUCTURES AND
h1=N(z1); h2=N(z2); h3=N(z3); h3v=N(z3v);
printerit printerit pontatroli por natrovi
b4=N[z4]; b5=N[z5]; rp=N[rx];
(* Se definen los puntos en el espacio
tridimensional para arafiaar al manipuladar ()
(nomensional para grancar el manipulador)
$\mathbf{n} = \{0, 0, 0\}$
punto i = {0,0,0},
punto2 = {0.0,base};
sunta 2 - sunta 2 + hd.
puntos = puntoz + b1,
punto4 = punto3 + b2;
number a municipal stability
puntos = punto4 + bs,
punto6 = punto5 + b3v
punto r = punto + bo;
punto8 = punto6 - h5
punto9 = punto7 + b4;
punto10 = punto8 + b4
punto to = puntoo + ber
punto11 = punto2 + rp;
(* Se agrega a la gráfica base la posición del
elemento terminal como un nunto *)
elemento terminar como un punto)
a2=/Graphice3DI/PointSize(0.01)
gz-/Graphicaop//romorzato.org
Point(punto11), RGBColor(0,1,0))], a2);
g3=Graphics3D[{
(* Base *)
(AbsoluteThickness[20] BCBColor(0.1.0)
(Ansolute Luickness[30], AGBCOloi[0, 1,0],
Line (punto1.punto2)]).
(* Eelahán 1 *)
AbsoluteThicknessi25).RGBColori14.01.
Linellounto2 nunto2111
Line[[puntoz,puntos]]],
(* Eslabón 2 *)
AbsoluteThickness[20] DCBColor(0, 6.4)
Line (punto3.punto4)]).
(* Esiabon 3 tijo *)
{ Estadon 3 mjo "} {AbsoluteThickness[15].RGBColor[14.0].
(*Esiabon 3 tijo*) (AbsoluteThickness[15],RGBColor[1,4,0],
(* Eslabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]},
(* Eslabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]}, (* Eslabón 3 variable*)
(* Esiabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]}, (* Esiabón 3 variable*) (AbsoluteThickness[2],BCBColor[0,0,1])
(* Esiabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]], (* Esiabón 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1],
(* Esiabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]}, (* Esiabón 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]}.
<pre>(* Eslabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]}, (* Eslabón 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]}, (f Diase t)</pre>
(* Eslabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]], (* Eslabón 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]], (* Pinza *)
(* Esiabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]}, (* Esiabón 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]}, (* Pinza *) {AbsoluteThickness[3],RGBColor[0,0,0].
<pre>(* Estabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]}, (* Estabón 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]}, (* Pinza *) {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto6}]]}</pre>
<pre>(* Eslabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]], (* Eslabón 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]], (* Pinza *) {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]],</pre>
<pre>(* Estabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]}, (* Estabon 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]}, (* Pinza *) {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]}, {AbsoluteThickness[3],RGBColor[0,0,0], AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8,punto8,punto7,punto8,punto8,punto7,punto8,punto7,punto8,punto8,punto7,punto8,punt</pre>
<pre>(* Eslabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]}, (* Eslabón 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]}, (* Pinza *) {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]}, {AbsoluteThickness[3],RGBColor[0,0,0], {Absol</pre>
<pre>(* Eslabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]], (* Eslabón 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]], (* Pinza *) {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punt09}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punt09}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punt07,punt09}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punt07,punt09}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punt07,punt09}]], {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punt07,punt09}]], {AbsoluteThickness[3],RGBColor[0,0,0], {Absol</pre>
<pre>(* Esiabon 3 Tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]}, (* Esiabón 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]}, (* Pinza *) {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]}, {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]}, {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]}, {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]}, {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto9}]}, {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,pu</pre>
<pre>(* Estabon 3 Tijo *)</pre>
<pre>(* Estabon 3 tijo *) {AbsoluteThickness[15],RGBColor[1,.4,0], Line[{punto4,punto5}]}, (* Estabon 3 variable*) {AbsoluteThickness[8],RGBColor[0,0,1], Line[{punto5,punto6}]}, (* Pinza *) {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]}, {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]}, {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto8}]}, {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto7,punto9}]}, {AbsoluteThickness[3],RGBColor[0,0,0], Line[{punto8,punto10}]} }];</pre>

SIMULACION COMPUTACIONAL DE LA CINEMATICA





CAPITULO V

5. EVASION DE OBSTACULOS EN LA CINEMATICA INVERSA (Robot de tres grados de libertad).

Finalmente, en este capitulo, mostraremos una aplicación de la modelación con rotaciones y reflexiones del robot de tres grados de libertad (Figura 5.1) que se presentó en el capítulo 2. La aplicación consiste en lograr que el manipulador evada un obstáculo virtual en el proceso de cinemática inversa. Como sabemos que la diferencia entre los dos modelos obtenidos para el robot en cuestión es, principalmente, que al resolver la cinemática inversa para una misma posición del elemento terminal, se obtienen dos diferentes configuraciones de los eslabones del manipulador, utilizaremos esta propiedad con el fin de lograr la evasión del obstáculo mencionado. Es decir, si el obstáculo impide que el manipulador se posicione en un punto determinado, cuando se usa el modelo con rotaciones, se seleccionará el modelo (rotaciones y reflexiones) para posicionar el manipulador en el punto deseado, sin impactar el obstáculo. A continuación presentamos los sistemas de ecuaciones que se deberán resolver, para determinar la cinemática inversa del manipulador de tres grados de libertad.



Figura 5.1. Configuración del manipulador de tres grados de libertad

Ecuaciones para la configuración no deformada (única). La cinemática inversa para la configuración no deformada consiste en calcular los parámetros $P=\{P_0, P_1, 0, 0\}$ y $Q=\{Q_0, Q_1, 0, 0\}$ dada la posición del elemento terminal $R_p = \{0, R_py, R_pz\}$ (Ecuaciones (3.11), (3.12)), que satisfagan el siguiente sistema de ecuaciones.

$$2^{*}((A2^{*}A3 + A1^{*}A4) *L_{3} - A1^{*}L_{2}) = R_{p}y$$

$$L_{1} - A2^{*}L_{2} + (A2^{*}A4 - 4^{*}A1^{*}A3)*L_{3} = R_{p}z$$

$$P_{0}^{2} + P_{1}^{2} = 1$$

$$Q_{0}^{2} + Q_{1}^{2} = 1$$
(5.1)

Siendo:

$$A1 = P_0^* P_1; \quad A2 = 1 - 2^* P_0^2;$$
(5.2)

$$A3 = Q_0^* Q_1; \quad A4 = 1 - 2^* Q_0^2;$$

Ecuaciones para la configuración deformada (rotaciones). La cinemática inversa para la configuración deformada, con el modelo con puras rotaciones(Ecuaciones (3.41), (3.42), (3.43)), consiste en que, dada la posición del elemento terminal $r_p = \{r_px_i, r_py, r_pz_i\}$, calcular los parámetros r1₀, r1₃, r2₀, r2₁, r3₀, r3₁, que satisfagan el siguiente sistema de ecuaciones.

$$r_{p}x = -4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}L_{3})$$

$$r_{p}y = -2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}L_{3})$$

$$r_{p}z = L_{1} + B1^{*}L_{2} + C2^{*}L_{3} \qquad (5.3)$$

$$r1_{0}^{2} + r1_{3}^{2} = 1$$

$$r2_{0}^{2} + r2_{1}^{2} = 1$$

$$r3_{0}^{2} + r3_{1}^{2} = 1$$

Siendo:

A1 =
$$P_0^*P_1$$
; A2 = $1-2^*P_0^2$;
A3 = $Q_0^*Q_1$; A4 = $1-2^*Q_0^2$;
A5 = $r1_0^*r1_3$; A6 = $1-2^*r1_0^2$;
A7 = $r2_0^*r2_1$; A8 = $1-2^*r2_0^2$;
A9 = $r3_0^*r3_1$; A10 = $1-2^*r3_0^2$; (5.4)
B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8;
B3 = A10*A8 - 4*A1*A7; B4 = A10*A7 + A8*A9;
B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4;
C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6;

Ecuacionea para la configuración deformada (rotaciones y reflexiones). La cinemática inversa para la configuración deformada, utilizando el modelo con rotaciones y reflexiones (Ecuaciones (3.59), (3.60), (3.61)), consiste en que, dada la posición del elemento terminal $r_p = \{r_px, r_py, r_pz,\}$, se deben determinar los parámetros R1₀, R1₃, R2₀, R2₁, R3₀, R3₁, que satisfagan el siguiente sistema de ecuaciones.

$$r_{p}x = 4^{*}A5^{*}(B2^{*}L_{2} + C1^{*}L_{3})$$

$$r_{p}y = 2^{*}A6^{*}(B2^{*}L_{2} + C1^{*}L_{3})$$

$$r_{p}z = L_{1} - B1^{*}L_{2} + C2^{*}L_{3} \qquad (5.5)$$

$$R1_{0}^{2} + R1_{3}^{2} = 1$$

$$R2_{0}^{2} + R2_{1}^{2} = 1$$

$$R3_{0}^{2} + R3_{1}^{2} = 1$$

Siendo:		
A1 = P.*P.;	A2 = 1-2"P.";	
A3 = Q. Q.;	A4 = 1-2"Q.";	
A5 = R1,"R1,;	A6 = 1-2"R1,2;	
A7 = R2, R2,;	A8 = 1-2*R2,2;	
A9 = R3,*R3,;	A10 = 1-2"R3,2;	(5.6)
B1 = A2*A8 + 4	*A1*A7; B2 = A1*A8 - A2*A7;	. ,
B3 = A10*A8 +	4*A7*A9; B4 = A10*A7 - A8*A9;	
85 = A2*A3 + A	1*A4; B6 = A2*A4 - 4*A1*A3;	
C1 = 84*86 - 83	3*85; C2 = 83*86 + 4*84*85;	

Lo anterior nos permite ubicar el elemento terminal en el punto {r_px, r_py, r_pz} dentro de su espacio de trabajo con las dos posibles configuraciones del manipulador, ahora si deseamos que el elemento terminal siga una trayectoria determinada, una de las cosas que podemos hacer es discretizar dicha trayectoria en varios puntos a lo largo de ella y determinar los parámetros de rotación de los eslabones del manipulador para cada punto, resolviendo los sistema de ecuaciones anteriores, lo que nos permitirá ubicar el elemento terminal sobre dicha secuencia de puntos.. A continuación se presenta el programa desarrollado en Mathematica[®] para mostrar, mediante la simulación computacional, el efecto de la evasión de obstáculos que se quiere mostrar.

En la primera parte del programa se define la transformación rotación de un vector tridimensional y de un Quaternión, y de algunas otras funciones necesarias que se muestran a continuación.	Definición de las Transformaciones Rotación y Reflexión.
Transformación de un Quaternión en un vector tridimensional.	(* Se apagan algunos mensajes del sistema *)
q = {q₀, q₁, q₂, q₃} ∈ Q	Off[General∷spell1] Off[General∷spell]
Tv[q] = {q ₁ , q ₂ , q ₃ }.	(* Se define la Transformación de un Quaternión a un vector tridimensional Tv[] *)
Transformación vectorial inversa.	T∨[q_]:={q[[2]],q[[3]],q[[4]]};
$v = \{v_1, v_2, v_3\} \in \mathfrak{R}^3$	(* Se define la Transformación inversa de Tv[] *)
Tvinv[v] = $\{0, v_1, v_2, v_3\}$.	Tvlnv[v_]:={0,v[[1]],v[[2]],v[[3]]};
Transformación de un Quaternión en un Quaternión vectorial.	(* Se define la Transformación de un Quaternión a Quaternión vectorial *)
TvQ[q] = {0,q ₁ , q ₂ , q ₃ }.	TvQ[q_]:={0,q[[2]],q[[3]],q[[4]]};
Producto de dos Quaterniones.	(* Se define la Multiplicación entre dos Quaterniones
$p = \{p_0, p_1, p_2, p_3\}, q = \{q_0, q_1, q_2, q_3\} \in Q$	•)
ProductoQ[p,q] = {p ₀ *q ₀ - p ₁ *q ₁ - p ₂ *q ₂ - p ₃ *q ₃ , p ₀ *q ₁ + p ₁ *q ₀ + p ₂ *q ₃ - p ₃ *q ₃ , p ₀ *q ₂ - p ₁ *q ₃ + p ₂ *q ₀ + p ₃ *q ₁ , p ₀ *q ₃ + p ₁ *q ₂ - p ₂ *q ₁ + p ₃ *q ₀ }	$\begin{array}{l} ProductoQ[p_,q_]:= \\ \{p[[1]]^q[[1]] \ - \ p[[2]]^q[[2]] \ - \ p[[3]]^q[[3]] \ - \ p[[4]]^q[[4]], \\ p[[1]]^q[[2]] \ + \ p[[2]]^q[[1]] \ + \ p[[3]]^q[[4]] \ - \ p[[4]]^q[[3]], \\ p[[1]]^q[[3]] \ - \ p[[2]]^q[[4]] \ + \ p[[3])^q[[1]] \ + \ p[[4])^q[[2]], \\ p[[1]]^q[[4]] \ + \ p[[2]]^q[[3]] \ - \ p[[3])^q[[2]] \ + \ p[[4])^q[[1]]; \end{array}$

Conjugado de un Ouetornión	(* Ca define al caniurada de un Ousternién *)
Conjugado de un Quaternion.	(Se define el conjugado de un Quaternion -)
ConjugadoQ[p] = { $p_0, -p_1, -p_2, -p_3$ }	ConjugadoQ[p_]:={p[[1]]p[[2]]p[[3]]p[[4]]};
Rotación de un vector v ∈ ℜ³ con un	
Quaternión p siendo p = 1 Según la	(* Se define la Transformación Rotación de un
definición (2.13)	vector tridimensional *)
$\rho(\mathbf{p},\mathbf{q}) = \frac{1}{\ \mathbf{p}\ ^2} \bullet \left(\mathbf{p}^*\mathbf{q}^*\overline{\mathbf{p}}\right), \forall \mathbf{q} \in \mathbf{Q}$	Rotacion[p_,v_]:≖Tv[ProductoQ[p, ProductoQ[TvInv[v], ConjugadoQ[p]]]];
 -	(* Se define la Transformación Rotación de un
Rotacion[p,v] = Tv[p * Tvinv[v] * p]	Quaternión *)
Rotación de a e O con un Quaternión n	RotacionQ[p_,q_]:={q[[1]],0,0,0} +
siendo n - 1. De souerdo con la definición	TvQ[ProductoQ[p,ProductoQ[q,ConjugadoQ[p]]]];
de q no cambia al rotar cuando $\ \mathbf{p}\ = 1$.	(* Se define la Transformación Reflexión de un vector tridimensional *)
	Reflexionin a 1= TviConlugadoQiProductoQin
RotacionQ[p,q] = $\{q_0,0,0,0\} + 1VQ[p,q, b]$	ProductoQ[TvInv[q],ConjugadoQ[p]]]]];
Reflexión de un vector v e 93° con un	
Quaternión n siendo n = 1 Según la	(* Se define la Transformación Reflexión de un
definición /2 27)	Quaternión *)
() 1 ()	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQIConiugadoQIProductoQIp.
$\mathbf{R}(\mathbf{p},\mathbf{q}) = \frac{1}{\ \mathbf{p}\ ^2} \bullet \left(\overline{\mathbf{p} \cdot \mathbf{q} \cdot \overline{\mathbf{p}}} \right), \forall \mathbf{q} \in \mathbf{Q}$	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$\mathbf{R}(\mathbf{p},\mathbf{q}) = \frac{1}{\left\ \mathbf{p}\right\ ^2} \bullet \left(\overline{\mathbf{p} \cdot \mathbf{q} \cdot \overline{\mathbf{p}}}\right), \forall \mathbf{q} \in \mathbf{Q}$	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ \left(\overline{p \cdot q \cdot \overline{p}}\right), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p \cdot Tvinv[v] \cdot \overline{p}}$]	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \cdot q \cdot \overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p \cdot TvInv[v] \cdot \overline{p}}$] Reflexión de q $\in Q$ con un Quaternión p	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \cdot q \cdot \overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p \cdot TvInv[v] \cdot \overline{p}}$] Reflexión de q $\in Q$ con un Quaternión p.	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \circ q \circ \overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p \circ Tv lnv[v] \circ \overline{p}}$] Reflexión de q \in Q con un Quaternión p, siendo $\ p\ = 1$. De acuerdo con la definición	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \cdot q \cdot \overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p \cdot Tvinv[v] \cdot \overline{p}}$] Reflexión de q \in Q con un Quaternión p. siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de a caracter aucardo $\ n\ = 4$	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \cdot q \cdot \overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p \cdot Tvinv[v] \cdot \overline{p}}$] Reflexión de q \in Q con un Quaternión p, siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \circ q \circ \overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p \circ Tvinv[v] \circ \overline{p}}$] Reflexión de q \in Q con un Quaternión p, siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p q] =	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \circ q \circ \overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p \circ Tvinv[v] \circ \overline{p}}$] Reflexión de q \in Q con un Quaternión p, siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p,q] =	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p^*q^*\overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p^*Tvinv[v]^*\overline{p}}$] Reflexión de q \in Q con un Quaternión p, siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p,q] = $\{q_0,0,0,0\} + TvQ[\overline{p^*q^*\overline{p}}]$	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \circ q \circ \overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p \circ TvInv[v] \circ \overline{p}}$] Reflexión de q \in Q con un Quaternión p, siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p,q] = {q ₀ ,0,0,0} + TvQ[$\overline{p \circ q \circ \overline{p}}$] Se realizan definiciones iniciales necesarias,	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]];
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p^*q^*\overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p^*TvInv[v]^*\overline{p}}$] Reflexión de q \in Q con un Quaternión p, siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p,q] = {q ₀ ,0,0,0} + TvQ[$\overline{p^*q^*\overline{p}}$] Se realizan definiciones iniciales necesarias, como son la base canónica de \Re^3 y las bases	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]]; Definiciones iniciales
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p^*q^*\overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p^*Tvlnv[v]^*\overline{p}}$] Reflexión de q \in Q con un Quaternión p. siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p,q] = $\{q_0,0,0,0\} + TvQ[\overline{p^*q^*\overline{p}}]$ Se realizan definiciones iniciales necesarias, como son la base canónica de \Re^3 y las bases que servirán como bases locales en los	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]]; Definiciones iniciales
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p^*q^*\overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p^*Tvlnv[v]^*\overline{p}}$] Reflexión de q \in Q con un Quaternión p. siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p,q] = $(q_{0},0,0,0) + TvQ[p^*q^*\overline{p}]$ Se realizan definiciones iniciales necesarias, como son la base canónica de \Re^3 y las bases que servirán como bases locales en los eslabones. También se define la forma local de leo Queterinere que con computitiva como	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]]; Definiciones iniciales (* Base canónica de R3*) e1 = (1 0 0); e2 = (0 1 0); e3 = (0 0 1);
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \cdot q \cdot \overline{p}}), \forall q \in Q$ Reflexion[p,v] = $Tv[\overline{p \cdot TvInv[v] \cdot \overline{p}}]$ Reflexion de q \in Q con un Quaternión p, siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p,q] = $(q_0,0,0,0) + TvQ[\overline{p \cdot q \cdot \overline{p}}]$ Se realizan definiciones iniciales necesarias, como son la base canónica de \Re^3 y las bases que servirán como bases locales en los eslabones. También se define la forma local de los Quaterniones que nos permitirán rotar, tanto en la posición no deformada como en la	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]]; Definiciones iniciales (* Base canónica de R3*) e1 = {1,0,0}; e2 = {0,1,0}; e3 = {0,0,1};
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \circ q \circ \overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[p ° Tvinv[v] ° \overline{p}] Reflexión de q \in Q con un Quaternión p, siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p,q] = $\{q_0,0,0,0\} + TvQ[p \circ q \circ \overline{p}]$ Se realizan definiciones iniciales necesarias, como son la base canónica de \Re^3 y las bases que servirán como bases locales en los eslabones. También se define la forma local de los Quaterniones que nos permitirán rotar, tanto en la posición no deformada como en la deformada.	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]]; Definiciones iniciales (* Base canónica de R3*) e1 = {1,0,0}; e2 = {0,1,0}; e3 = {0,0,1}; (* Bases locales de los eslabones *)
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p \cdot q \cdot \overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p \cdot TvInv[v] \cdot \overline{p}}$] Reflexión de q \in Q con un Quaternión p, siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p,q] = $\{q_0,0,0,0\} + TvQ[\overline{p \cdot q \cdot \overline{p}}]$ Se realizan definiciones iniciales necesarias, como son la base canónica de \Re^3 y las bases que servirán como bases locales en los eslabones. También se define la forma local de los Quaterniones que nos permitirán rotar, tanto en la posición no deformada como en la deformada.	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]]; Definiciones iniciales (* Base canónica de R3*) e1 = {1,0,0}; e2 = {0,1,0}; e3 = {0,0,1}; (* Bases locales de los eslabones *) (* Eslabón 1 *)
$R(p,q) = \frac{1}{\ p\ ^2} \circ (\overline{p^*q^*\overline{p}}), \forall q \in Q$ Reflexion[p,v] = Tv[$\overline{p^*Tvlnv[v]^*\overline{p}}$] Reflexión de q \in Q con un Quaternión p. siendo $\ p\ = 1$. De acuerdo con la definición (2.27) y considerando que el primer elemento de q no cambia al rotar cuando $\ p\ = 1$ ReflexionQ[p,q] = $\{q_0,0,0,0\} + TvQ[\overline{p^*q^*\overline{p}}]$ Se realizan definiciones iniciales necesarias, como son la base canónica de \Re^3 y las bases que servirán como bases locales en los eslabones. También se define la forma local de los Quaterniones que nos permitirán rotar, tanto en la posición no deformada como en la deformada.	ReflexionQ[p_,q_]:={q[[1]],0,0,0} + TvQ[ConjugadoQ[ProductoQ[p, ProductoQ[q,ConjugadoQ[p]]]]]; Definiciones iniciales (* Base canónica de R3*) e1 = {1,0,0}; e2 = {0,1,0}; e3 = {0,0,1}; (* Bases locales de los eslabones *) (* Eslabón 1 *) e1 = {1,0,0}; e2 = {0,1,0}; e3 = {0,0,1};

(* Eslabón 2 *) $f1 = \{1,0,0\}; f2 = \{0,1,0\}; f3 = \{0,0,1\};$ (* Eslabón 3 *) $q1 = \{1,0,0\}; q2 = \{0,1,0\}; q3 = \{0,0,1\};$ ("Quaterniones para rotar a posición no deformada") P = {P0,P1,0,0}; (* Eslabón 2 *) $Q = \{Q0, Q1, 0, 0\}; (* Eslabón 3 *)$ (* Quaterniones para rotar a posición deformada *) (* Para rotaciones *) r1 = {r10,0,0,r13}; (* Eslabón 1 *) r2 = {r20,r21,0,0}; (* Eslabón 2 *) r3 = {r30,r31,0,0}; (* Eslabón 3 *) (* Para rotaciones y reflexiones *) R1 = {R10,0,0,R13}; (* Eslabón 1 *) R2 = {R20,R21,0,0}; (* Eslabón 2 *) R3 = {R30,R31,0,0}; (* Eslabón 3 *) MODELADO DE LA CONFIGURACION NO En esta sección del programa se realiza la DEFORMADA Modelación de la configuración no deformada del manipulador. Como la forma de las bases locales de los eslabones, en la configuración Modelo para la configuración no deformada no deformada, se obtuvo en el capitulo anterior, entonces para facilitar la obtención (* Se inicializan variables *) ClearAll[P0,P1,Q0,Q1,L1,L2,L3,b1,b2,b3]; del las ecuaciones del manipulador, simplemente obtenemos de manera directa la forma de las bases locales, que son: (* Se definen variables para las normas unitarias de los Quaterniones P y Q *) $\mathbf{f}_i^{\mathrm{I}} = \rho(\mathbf{P}, \mathbf{f}_i) \qquad (\text{Eslabón 2})$ normas1 ={P1->(1-P0^2)^(1/2),Q1->(1-0^2)^(1/2)}; $\mathbf{g}_{i}^{II} = \rho((\mathbf{P}^{*}\mathbf{Q}), \mathbf{g}_{i})$ (Eslabón 3) normas2 ={ $(1-P0^2)^{(1/2)} > P1, (1-Q0^2)^{(1/2)} > Q1$ }; (* Se obtienen los productos de Quaterniones Para lo anterior es necesario definir las necesarios para rotar las bases *) normas unitarias de los Quaterniones P y Q, las cuales se almacenan en las variables PQ = ProductoQ[P,Q];"normas1" y "normas2", para aplicarlas en las ecuaciones resultantes de las rotaciones. (* Se obtiene la forma de las bases locales de los eslabones en la configuración no deformada *) Las variables que se utilizan para las bases locales de los eslabones son: (* Eslabón 1 *) $e1 = \{1,0,0\}; e2 = \{0,1,0\}; e3 = \{0,0,1\};$ ei = Primer eslabón. fj = Segundo eslabón. (* Eslabón 2 *) gi = Tercer eslabón. f11 = Expand[Expand[Rotacion[P,f1]]//.normas1] //.normas2: f21 = Expand[Expand[Rotacion[P,f2]]//.normas1] //.normas2; f31 = Expand[Expand[Rotacion[P,f3]]//.normas1]

	ll a same a O
	//.normasz;
	(* Eslabon 3 *)
	g12 = Expand[Expand[Rotacion[PQ,g1]]//.normas1]
	//.normas2;
	g22 = Expand[Expand[Rotacion[PQ,g2]]//.normas1]
	//.normas2;
	g32 = Expand/Expand/Rotacion/PQ.g311//.normas11
	// normas2
Los estabones del maninulador, y la naminida	/* Vectores que definen los estaboras dal
dei elemento terminal con identificados en la	manipulador *)
der elemento terminal son identificados, en la	(manipuladon)
posición no deformada, por las variables	
siguientes:	D1 = L1*83; (* ESIADON 1 *)
	b2 = L2131; (* Eslabon 2 *)
b1 = Esiabón 1.	b3 = L3*g32; (* Eslabón 3 *)
b2 = Eslabón 2.	
b3 = Eslabón 3.	(* Constantes que se repiten y que servirán para
	simplificar las ecuaciones y optimizar operaciones *)
Bp = b1+b2+b3;	
	constantes = $(P0*P1>A1 P0^2>(1-A2))/2$
Do = Dosición del elemento terminal	00+01-543 0042 5/4 44/2)
ry - rosición del elemento terminal.	<u>uuuir~~~, uu~~~(1~~4)/2},</u>
Las ecuaciones se simplifican utilizando las	DZ = DZ//.constantes;
siguientes constantes	b3 = Expand[Expand[b3//.constantes]//.constantes];
A1 = P0"P1; A2 = 1 - 2"P0^2;	(* vector que define posición del elemento terminal*)
A3 = Q0"Q1; A4 = 1 - 2"Q0^2;	
$A3 = Q0^{\circ}Q1; A4 = 1 - 2^{\circ}Q0^{2};$	Rp = b1+b2+b3;
A3 = Q0"Q1; A4 = 1 - 2"Q0^2;	Rp = b1+b2+b3;
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada.
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada.
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1]
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1]
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *)
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *)
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir,	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *)
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación	Rp = b1+b2+b3;Ecuaciones optimizadas para la posición no deformada.ClearAll[P0,P1,Q0,Q1](* Ecuaciones optimizadas *)b1 = {0, 0, L1}; (* Esiabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Esiabón 2 *)
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *)
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tras grados de libertad	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3):
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Esiabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Esiabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Esiabón 3 *) (A2*A4 - 4*A1*A3)*L3};
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3};
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *)
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) Rp = (0, 21/(A2*A2 + A4*A4)*L2, A4*L2)
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal,	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2),
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente:	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3};
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente:	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* A1*A3)*L3;
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: $Rp = \{0, Rpy, Rpz\}.$	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *)
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: $Rp = \{0, Rpy, Rpz\}.$	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2;
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: $Rp = \{0, Rpy, Rpz\}.$	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2;
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2;
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *)
A3 = Q0°Q1; A4 = 1 - 2°Q0^2; Es esta parte se muestran las ecuaciones obtenidas en la sección anterior, las cuales se utilizarán para determinar los parámetros P y Q, que definen la configuración no deformada en una posición dada del elemento terminal, que será de donde arranque la simulación de la cinemática directa del manipulador en cuestión. Es decir, antes de generar una simulación computacional de la cinemática directa del manipulador de tres grados de libertad, debemos determinar la configuración no deformada realizando un proceso de cinemática inversa para calcular P y Q para una posición conocida del elemento terminal, la cual debe tener la forma siguiente: Rp = {0,Rpy, Rpz}.	Rp = b1+b2+b3; Ecuaciones optimizadas para la posición no deformada. ClearAll[P0,P1,Q0,Q1] (* Ecuaciones optimizadas *) b1 = {0, 0, L1}; (* Eslabón 1 *) b2 = {0, -2*A1*L2, -A2*L2}; (* Eslabón 2 *) b3 = {0, 2*(A2*A3 + A1*A4)*L3, (* Eslabón 3 *) (A2*A4 - 4*A1*A3)*L3}; (* Posición el elemento terminal *) Rp = {0, 2*((A2*A3 + A1*A4)*L3 - A1*L2), L1 - A2*L2 + (A2*A4 - 4*A1*A3)*L3}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1 - 2*P0^2; A3 = Q0*Q1; A4 = 1 - 2*Q0^2; (* Longitud de los eslabones del manipulador *)

Se resuelve el sistema de ecuaciones mostrado en (5.1), para P0,P1,Q0,Q1,	Cinemática Inversa para determinar la Configuración no Deformada
se puede consulta en el Apéndice, al igual que todas las funciones que se utilizan en este trabajo.	(* Conociendo la posición del elemento terminal se determinan los Quaterniones "P" y "Q" correspondientes *)
alguno de ellos está muy cercano a cero	ClearAllIPO P1 00 011
finalmente se almacenan en las variables	(* Se da la nosición del elemento terminal *)
correspondientes (P0, P1, Q0, Q1) para	
poder ser utilizados posteriormente, además.	posicion = {0, 0, 15};
se imprimen dichos valores.	
	(* Se determinan "P" y "Q" utilizando los siguiente valores iniciales *)
	iP0 = .7071; iP1 = .7071; iQ0 = .866; iQ1 = .5;
	(* Se almacena la solución en la variable sol y se utiliza la función FindRoot para determinar la solución del sistema de ecuaciones *)
	sol = FindRoot[{Rp2 == posicion[[2]],Rp3 == posicion[[3]], P0^2+P1^2 == 1, Q0^2+Q1^2 == 1}, {P0,iP0}, {P1,iP1}, {Q0,iQ0}, {Q1,iQ1}, MaxIterations->50];
	(* Se asignan los valores de la solución a las variables correspondientes *)
	P0=P0/.sol; P1=P1/.sol; Q0=Q0/.sol; Q1=Q1/.sol;
	(* Se redondean los valores de la solución *)
	lf[Abs[P0]==1,P1=0,lf[Abs[P1]==1,P0=0]]; lf[Chop[Abs[Q0]-1]==0,Q0=1;Q1=0, lf[Chop[Abs[Q1]-1]==0,Q0=0;Q1=1]];
	(* Se imprimen los resultados *)
	Print["P0 = ",P0 ," ", "P1 = ", P1, " Q0 = ", Q0, " ", "Q1 = ", Q1]
De la misma forma que se realizó la modelación de la configuración no	MODELADO DE LA CONFIGURACION DEFORMADA
deformada, ahora se presenta el cálculo de la forma final de las bases locales de los esiabones, las cuales se obtuvieron en el capitulo anterior y son:	Modelo para la configuración deformada con rotaciones.
	ClearAll[r10,r13,r20,r21,r30,r31,L1,L2,L3];
$\mathbf{e}_{j}^{1} = \rho(\mathbf{r1}, \mathbf{e}_{j})$	(* Posición no deformada conocida *)
$f_{j}^{'''} = \rho(r1^{*}r2^{*}P, f_{j})$	$P = \{P0, P1, 0, 0\}; Q = \{Q0, Q1, 0, 0\};$

$g_{1}^{V} = o(r1^{*}r2^{*}r3^{*}P^{*}Q, g_{1})$	(* Quaterniones para rotar a la configuración
	deformada *)
Tembite en este seco es definen los normes	r1 = {r10,0,0,r13}; (* Rotación Eslabón 1*)
Tambien en este caso, se definen las normas	r2 = {r20,r21,0,0}; (* Rotación Eslabón 2*)
unitarias de los Quaterniones utilizados,	r3 = {r30,r31,0,0}; (* Rotación Eslabón 3*)
almacenandolas en las variables normass	
y "normas4", que incluyen las normas	(* Normas unitarias de los Quaterniones *)
unitarias de los Quaterniones	A DAVER AND RECEIPTION OF A COMPANY OF A COMPANY OF
correspondientes a la configuración no	normas3 = Flatten[{normas1,
deformada.	r13->(1-r10^2)^(1/2), r21->(1-r20^2)^(1/2),
	r31->(1-r30^2)^(1/2), r43->(1-r40^2)^(1/2)}];
	normas4 = Flatten[{normas2,
	(1-r10^2)^(1/2)->r13, (1-r20^2)^(1/2)->r21,
	(1-r30^2)^(1/2)->r31, (1-r40^2)^(1/2)->r43}];
	(* Se obtienen los productos de Quaterniones
	necesarios nara rotar las bases *)
	r1r2P = ProductoQ[r1.ProductoQ[r2.P]];
	r1r2r3PQ = ProductoQ[r1, ProductoQ[r2,
	ProductoQ[r3,ProductoQ[P,Q]]]];
	(° Se obtiene le forme fine) de les bases locales de
	los eslabones *)
	(* Base local del eslabón 1 *)
	e11 = Expand[Expand[Rotacion[r1,e1]]//.normas3]
	//.normas4;
	e21 = Expand[Expand[Rotacion[r1,e2]]//.normas3]
	//.normase;
·	//.normas4;
	(* Base local del eslabón 2 *)
	f13 = Expand/Expand/Rotacion/r1r2P.f111//.normas31
	//.normas4:
	f23 = Expand/Expand/Rotacion/r1r2P.f2]]//.normas3]
	//.normas4:
	f33 = Expand/Expand/Rotacion/r1r2P.f3]]//.normas3]
	//.normas4;
	(* Base local del esiabón 3 *)
	a15 = Expand/Expand/Rotacion/r1r2r3PO a111
	// normas31// normas4:
	a25 = Expand/Expand/Rotacion/r1/2/3PO a211
	// normas3// normas4:
	a35 = Expand Expand Rotacion (r1r2r3DO a3)
	//.normas3]//.normas4;
Una vez obtenidas las bases locales en su	
rorma inal, se obterien las ecuaciones que	h1 = 11*e31 /* EalebAn 1 *\
representan los eslabones del manipulador y	$b_1 = 12^{123}$ (* Felabán 2 *)
a posición del elemento terminal como:	$b_2 = L_2 (35, (-5) = 0.000 + 2.000 $

$b_1 = L_1^* e_1^1 = L_1^* e_3^* (Estabon 1)$	(* Constantes que se repiten y que servirán para
$b_2 = L_2^* f_3^{(1)} = L_2^* f_3^{(3)};$ (Eslabón 2)	simplificar las ecuaciones y optimizar operaciones *)
$b_{1} = 1^{+} a_{1}^{V} = 1^{+} a_{1}^{2} 5^{+} (Felebén 3)$	constantes = $(P0^{\circ}P1 - > A1 P0^{\circ}2 - > (1 - A2))/2$
$D_3 = C_3 B_3 = C_3 B_0 O_1 (COURDOULD)$	Q0*Q1->A3_Q0*2->(1-A4)/2
(Decisión del clomente terminel)	r10*r13->A5 r10^2->(1-A6)/2
(Posicion del elemento terminal)	r20*r21->A7_r20^2->(1-A8)/2
	r30*r31->A9_r30^2->(1-A10)/2
$r_p = D_1 + D_2 + D_3$	A2*A8->4*A1*A7+B1 A2*A7->B2-A1*A8
Fotos everencianos por cimplificados	A10*A8->4*A7*A9+B3 A10*A7->B4-A8*A9
Estas expresiones son simplificadas	A2*A4->4*A1*A3+B5, A2*A3->B6-A1*A4
sustituyendo por constantes a aquellos	B4*B5->C1 - B3*B6, B3*B5->C2+4*B4*B6);
diferentes ocuaciones como se puede	
chrenies ecuaciones, como se puede	b2 = Expand/b2//.constantes];
	b2 = Expand(Collect(b2, (L2, A5, A6))// constantes);
	b3 = Expand[b3//.constantes];
	b3 = Collect[Collect[Collect[b3, (L3, A5, A1, A2, A3, A4,
	A6}]//.constantes, {L3,A5,A6,B3,B4}]
	//.constantes.{L3.A5.A6.B3.B4}];
	b3 = Expand[b3//.constantes];
	(*Vector que define posición del elemento terminal *)
	rp = Factor[b1 + b2 + b3];
De la misma forma que se realizó la	Modelo para la configuración deformada con
modelación con puras rotaciones, ahora se	rotaciones y reflexiones.
modelación con puras rotaciones, ahora se determina la forma final de las bases locales	rotaciones y reflexiones.
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son:	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *)
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son:	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll/R10 R13 R20 R21 R30 R31 L1 L2 L3]
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $e_j^{T} = \rho(r1, e_j)$	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3];
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capítulo anterior y son: $e_j^{I} = \rho(r1, e_j)$ $f_i^{III} = R(r1 * r2 * P, f_j)$	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *)
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capítulo anterior y son: $e_j^{I} = \rho(r1, e_j)$ $f_j^{III} = R(r1^*r2^*P, f_j)$	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *)
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capítulo anterior y son: $e_j^{I} = \rho(r1, e_j)$ $f_j^{III} = R(r1 + r2 + P, f_j)$ $g_j^{V} = \rho(r1 + r2 + r3 + P + Q, g_j)$	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0};
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capítulo anterior y son: $e_j^{I} = \rho(r1, e_j)$ $f_j^{III} = R(r1^*r2^*P, f_j)$ $g_j^{V} = \rho(r1^*r2^*r3^*P^*Q, g_j)$	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0};
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $e_j^{I} = \rho(r1, e_j)$ $f_j^{III} = R(r1^*r2^*P, f_j)$ $g_j^{V} = \rho(r1^*r2^*r3^*P^*Q, g_j)$ También en este caso, se definen las normas	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración distances para rotar a la configuración
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $e_j^{I} = \rho(r1, e_j)$ $f_j^{III} = R(r1^*r2^*P, f_j)$ $g_j^{V} = \rho(r1^*r2^*r3^*P^*Q, g_j)$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados,	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) P = (P1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $\mathbf{r}_{j}^{III} = \rho(\mathbf{r1}, \mathbf{r}_{j})$ $\mathbf{f}_{j}^{III} = \mathbf{R}(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{f}_{j})$ $\mathbf{g}_{j}^{V} = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{g}_{j})$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3"	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {P24.0.0; (* Rotación Eslabón 1*)
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $\mathbf{r}_{j}^{I} = \rho(\mathbf{r1}, \mathbf{r}_{j})$ $\mathbf{f}_{j}^{III} = \mathbf{R}(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{f}_{j})$ $\mathbf{g}_{j}^{V} = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{g}_{j})$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3" y "normas4", que incluyen las normas	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {R20,R21,0,0}; (* Rotación Eslabón 2*) P = {P0,P1,0,0}; (* Rotación Eslabón 2*)
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $\mathbf{r}_{j}^{I} = \rho(\mathbf{r1}, \mathbf{r}_{j})$ $\mathbf{f}_{j}^{III} = \mathbf{R}(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{f}_{j})$ $\mathbf{g}_{j}^{V} = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{g}_{j})$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3" y "normas4", que incluyen las normas unitarias de los Quaterniones	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {R20,R21,0,0}; (* Rotación Eslabón 2*) R3 = {R30,R31,0,0}; (* Rotación Eslabón 3*)
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $\mathbf{r}_{j}^{I} = \rho(\mathbf{r1}, \mathbf{e}_{j})$ $\mathbf{f}_{j}^{III} = \mathbf{R}(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{f}_{j})$ $\mathbf{g}_{j}^{V} = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{g}_{j})$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3" y "normas4", que incluyen las normas unitarias de los Quaterniones correspondientes a la configuración no deformada.	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {R20,R21,0,0}; (* Rotación Eslabón 2*) R3 = {R30,R31,0,0}; (* Rotación Eslabón 3*) (* Normas unitarias de los Quaterniones *)
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los estabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $\mathbf{r}_{j}^{III} = \mathbf{R}(\mathbf{r1}^*\mathbf{r2}^*\mathbf{P}, \mathbf{f}_{j})$ $\mathbf{g}_{j}^{V} = \mathbf{p}(\mathbf{r1}^*\mathbf{r2}^*\mathbf{r3}^*\mathbf{P}^*\mathbf{Q}, \mathbf{g}_{j})$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3" y "normas4", que incluyen las normas unitarias de los Quaterniones correspondientes a la configuración no deformada.	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {R20,R21,0,0}; (* Rotación Eslabón 2*) R3 = {R30,R31,0,0}; (* Rotación Eslabón 3*) (* Normas unitarias de los Quaterniones *) normae5 = Elatten[/normas1
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los estabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $\mathbf{e}_{j}^{I} = \rho(\mathbf{r1}, \mathbf{e}_{j})$ $\mathbf{f}_{j}^{III} = \mathbf{R}(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{f}_{j})$ $\mathbf{g}_{j}^{V} = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{g}_{j})$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3" y "normas4", que incluyen las normas unitarias de los Quaterniones correspondientes a la configuración no deformada.	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {R20,R21,0,0}; (* Rotación Eslabón 2*) R3 = {R30,R31,0,0}; (* Rotación Eslabón 3*) (* Normas unitarias de los Quaterniones *) normas5 = Flatten[{normas1, R13->(1,R10^2)^(1/2)
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $e_j^I = \rho(r1, e_j)$ $f_j^{III} = R(r1^*r2^*P, f_j)$ $g_j^V = \rho(r1^*r2^*r3^*P^*Q, g_j)$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3" y "normas4", que incluyen las normas unitarias de los Quaterniones correspondientes a la configuración no deformada.	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {R20,R21,0,0}; (* Rotación Eslabón 2*) R3 = {R30,R31,0,0}; (* Rotación Eslabón 3*) (* Normas unitarias de los Quaterniones *) normas5 = Flatten[{normas1, R13->(1-R10^2)^(1/2), R21->(1-R20^2)^(1/2)
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $e_j^I = \rho(r1, e_j)$ $f_j^{III} = R(r1^*r2^*P, f_j)$ $g_j^V = \rho(r1^*r2^*r3^*P^*Q, g_j)$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3" y "normas4", que incluyen las normas unitarias de los Quaterniones correspondientes a la configuración no deformada.	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {R20,R21,0,0}; (* Rotación Eslabón 2*) R3 = {R30,R31,0,0}; (* Rotación Eslabón 3*) (* Normas unitarias de los Quaterniones *) normas5 = Flatten[{normas1, R13->(1-R10^2)^(1/2), R31->(1-R30^2)^(1/2)];
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $\mathbf{r}_{j}^{III} = \mathbf{r}(\mathbf{r1}, \mathbf{r}_{j})$ $\mathbf{g}_{j}^{V} = \mathbf{r}(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{g}_{j})$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3" y "normas4", que incluyen las normas unitarias de los Quaterniones correspondientes a la configuración no deformada.	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {R20,R21,0,0}; (* Rotación Eslabón 2*) R3 = {R30,R31,0,0}; (* Rotación Eslabón 3*) (* Normas unitarias de los Quaterniones *) normas5 = Flatten[{normas1, R13->(1-R10^2)^(1/2), R21->(1-R20^2)^(1/2)];
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $e_j^{I} = \rho(r1, e_j)$ $f_j^{III} = R(r1^*r2^*P, f_j)$ $g_j^{V} = \rho(r1^*r2^*r3^*P^*Q, g_j)$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3" y "normas4", que incluyen las normas unitarias de los Quaterniones correspondientes a la configuración no deformada.	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {R20,R21,0,0}; (* Rotación Eslabón 2*) R3 = {R30,R31,0,0}; (* Rotación Eslabón 3*) (* Normas unitarias de los Quaterniones *) normas5 = Flatten[{normas1, R13->(1-R10^2)^(1/2), R31->(1-R30^2)^(1/2)}]; normas6 = Flatten[{normas2,
modelación con puras rotaciones, ahora se determina la forma final de las bases locales de los eslabones, para la modelación con rotaciones y reflexiones, las cuales se obtuvieron en el capitulo anterior y son: $\mathbf{e}_{j}^{I} = \rho(\mathbf{r1}, \mathbf{e}_{j})$ $\mathbf{f}_{j}^{III} = \mathbf{R}(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{P}, \mathbf{f}_{j})$ $\mathbf{g}_{j}^{V} = \rho(\mathbf{r1}^{*}\mathbf{r2}^{*}\mathbf{r3}^{*}\mathbf{P}^{*}\mathbf{Q}, \mathbf{g}_{j})$ También en este caso, se definen las normas unitarias de los Quaterniones utilizados, almacenándolas en las variables "normas3" y "normas4", que incluyen las normas unitarias de los Quaterniones correspondientes a la configuración no deformada.	rotaciones y reflexiones. (* A las variables utilizadas exclusivamente para el modelo con rotaciones y las distinguiremos agregándoles al final una letra "R" *) ClearAll[R10,R13,R20,R21,R30,R31,L1,L2,L3]; (* Posición no deformada conocida *) P = {P0,P1,0,0}; Q = {Q0,Q1,0,0}; (* Quaterniones para rotar a la configuración deformada *) R1 = {R10,0,0,R13}; (* Rotación Eslabón 1*) R2 = {R20,R21,0,0}; (* Rotación Eslabón 2*) R3 = {R30,R31,0,0}; (* Rotación Eslabón 3*) (* Normas unitarias de los Quaterniones *) normas5 = Flatten[{normas1, R13->(1-R10^2)^(1/2), R31->(1-R30^2)^(1/2)]; normas6 = Flatten[{normas2, (1-R10^2)^(1/2)->R13,

	(1-R20^2)^(1/2)->R21, (1-R30^2)^(1/2)->R31}];
	(* Se obtienen los productos de Quaterniones
	necesarios para rotar las bases *)
	R1R2cP = ProductoQ[R1,ProductoQ]
	ConjugadoQ[R2],P]];
	R1R2cR3PQ = ProductoQ[R1,ProductoQ[
	ConjugadoQ[R2],ProductoQ[R3,ProductoQ[P,Q]]]];
	(* Se obtiene la forma final de las bases locales de
	(* Base local del eslabón 1 *)
	e11R = Expand[Expand[Rotacion[R1,e1]]//.normas5] //.normas6;
	e21R = Expand[Expand[Rotacion[R1,e2]]//.normas5] //.normas6;
	e31R = Expand[Expand[Rotacion[R1,e3]]//.normas5] //.normas6;
	(* Base local del eslabón 2 *)
	f13R = Expand[Expand[Reflexion[R1R2cP,f1]]
	//.normas5]//.normas6;
	f23R = Expand[Expand[Reflexion[R1R2cP,f2]]
	//.normas5]//.normas6;
	//.normas5]//.normas6;
	(* Base local del eslabón 3 *)
	g15R = Expand[Expand[Rotacion[R1R2cR3PQ,g1]]
	//.normas5]//.normas6;
	gzok = Expand[Expand[Kotacion[KTK2CK3PQ,gz]] // normae5]// normae6
	q35R = Expandi Expandi Rotacioni R1R2cR3PQ.q31
	//.normas5]//.normas6;
Una vez obtenidas las bases locales en su	(- vectores que definen los esiabones del
representan los eslabones del manipulador v	reflexiones *)
la posición del elemento terminal como:	b1R = L1*e31R; (* Eslabón 1 *)
	b2R = L2*f33R; (* Eslabón 2 *)
$b_1 R = L_1^* e_3^I = L_1^* e_3 I R;$ (Eslabón 1)	b3R ≖ L3*g35R; (* Eslabón 3 *)
$b_2 R = L_2^* f_3^{III} = L_2^* f_3 R;$ (Eslabón 2)	(* Constantes que se repiten en las ecuaciones y
$b_3R = L_3^* g_3^V = L_3^* g_35R;$ (Eslabón 3)	que servirán para simplificar las ecuaciones y optimizar operaciones *)
(Posición del elemento terminal)	
	$\frac{ \text{constantes} = \{\text{PU}^{-}\text{P1}^{-}\text{A1}, \text{PU}^{-}\text{2}^{-}\text{(1-A2)}/2, \\ 00^{\circ}\text{O1}^{\circ}\text{A3}, 0002^{\circ}\text{A1}/2, \\ 00^{\circ}\text{O1}^{\circ}\text{A3}, 0002^{\circ}\text{A1}/2, \\ 00^{\circ}\text{A1}^{\circ}\text{A2}/2, \\ 00^{\circ}\text{A1}^{\circ}\text{A2}/2, \\ 00^{\circ}\text{A1}^{\circ}\text{A2}/2, \\ 00^{\circ}\text{A1}^{\circ}\text{A2}/2, \\ 00^{\circ}\text{A1}/2, \\ 00^{\circ}\text{A1}/2$
$r_{p}R = b_{1}R + b_{2}R + b_{3}R$	R10*R13->A5R. R10*2->(1-A4//2,
Fetas expresiones con simplificadas	R20*R21->A7R, R20^2->(1-A8R)/2.
sustituvendo por constantes a aquellos	R30*R31->A9R, R30^2->(1-A10R)/2,
términos que se repiten dentro de las	A2*A8R->-4*A1*A7R+B1R,
	A1*A8R->B2R+A2*A7R,

diferentes ecuaciones, como se puede observar.	A10R*A8R->B3R-4*A7R*A9R, A10R*A7R->B4R+A8R*A9R, A2*A3->B5R-A1*A4, A2*A4->B6R+4*A1*A3, B4R*B6R->C1R+B3R*B5R, B3R*B6R->C2R-4*B4R*B5R}; b2R = Expand[Expand[b2R//.constantes] //.constantes]; b3R = Collect[Expand[b3R//.constantes], {L3,A1,A2,A3,A4}]; b3R = Collect[Expand[b3R//.constantes], {L3,B3R,B4R,A7R}]; b3R = Expand[Expand[b3R//.constantes], {L3,B3R,B4R,A7R}]; b3R = Expand[Expand[b3R//.constantes], {L3,B3R,B4R,A7R}]; b3R = Expand[Expand[b3R//.constantes], //.constantes]; (* Vector que define posición del elemento terminal*) rpR = Factor[b1R + b2R + b3R];
En esta parte se presenta el resultado de la obtenido de las secciones anteriores, es decir, las ecuaciones simplificadas del manipuladar así	Ecuacionea optimizadas para determinar configuración deformada
utilizadas en dichas ecuaciones. Esta sección es la que propiamente se utiliza para realizar	ClearAli(r10, r13, r20, r21, r30, r31, R10, R13, R20, R21, R30, R31);
manipulador de tres grados de libertad, ya que es la que contiene las ecuaciones ya	(* Con Rotaciones *)
simplificadas. Puede observarse, que además de las ecuaciones de los eslabones y de la posición del elemento terminal, se tiene una ecuación para e11 = e_1^I ; ésta sirve para mostrar el movimiento del primer eslabón, que al girar sobre su propio eje, requiere que se dibuje una de las bases	(* Eslabón 1 *) b1 = {0, 0, L1}; (* Eslabón 2 *) b2 = {-4*A5*B2*L2, -2*A6*B2*L2, B1*L2}; (* Eslabón 3 *) b3 = {-4*A5*C1*L3, -2*A6*C1*L3, C2*L3};
perpendiculares al él para ver claramente su movimiento.	(* Posición del elemento terminal *)
Como al realizar la simulación es necesario evaluar numéricamente las ecuaciones del manipulador, se genera un respaldo de	rp = {-4*A5*(B2*L2 + C1*L3), -2*A6*(B2*L2 + C1*L3), L1 + B1*L2 + C2*L3};
dichas ecuaciones en esta misma sección, almacenándolas en nuevas variables.	(* Componente 1 de la Base local del eslabón 1 (para dibujar la base) *)
	e11 = {-A6, 2*A5, 0};
	(* Con Reflexiones *)
	(* Eslabón 1 *) b1R = {0, 0, L1}; (* Eslabón 2 *) b2R = {4*A5R*B2R*L2, 2*A6R*B2R*L2, -B1R*L2}; (* Eslabón 3 *) b3R = {4*A5R*C1R*L3, 2*A6R*C1R*L3, C2R*L3};

rpR = {4*A5R*(B2R*L2 + C1R*L3), 2*A6R*(B2R*L2 + C1R*L3), L1 - B1R*L2 + C2R*L3); (* Componente 1 de la Base local del eslabón * (para dibujar la base) *) e11R = {-A6R, 2*A5R, 0}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1-2*P0^2; A3 = Q0*Q1; A4 = 1-2*Q0^2; A5 = r10*r13; A6 = 1-2*r10^2; A7 = r20*r21; A8 = 1-2*r30^2; B1 = A2*A6 + 4*A1*A7; B2 = A2*A7 + A1*A6; B3 = A10*A6 + 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6 A5R = R10*R13; A6R = 1-2*R10^2; A7R = R20*R21; A8R = 1-2*R20^2; A9R = R30*R31; A10R = 1-2*R30^2; B1R = A2*A8R + 4*A1*A7R; B2R = A10R*A7R - A8*A6R; B3R = A10R*A7R + A8*A6R; B3R = A10R*A7R + A8*A7R*A9; B4R = A10R*A7R - A8*A6R; B5R = A2*A4 + 4*1*A7; B2R = A10R*A7R - A8*A6R; B4R = A10R*A7R - A8*A6R; B4R = A10R*A7R - A8*A6R; B4R = A10R*A7R - A8*A6R; B4R = A10R*A7R + 4*B4*B6R; B4R = A10R*A7R + 4*B4*B6R; B4R = A10R*A7R + 4*B4*B6R; B4R = A10R*A7R + 4*B4*B6R; B4R = A2*A4 + 4*A1*A3; C1 = 64R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; C2R = B3R*B6R + 4*B4R		(* Posición del elemento terminal *)
rpR = {4'ASR'(B2R'L2 + C1R'L3), 2'AGR'(B2R'L2 + C1R'L3), L1 - B1R'L2 + C2R'L3); (* Componente 1 de la Base local del esiabón * (para dibujar la base) *) e11R = {-AGR, 2*ASR, 0}; (* Constantes de las ecuaciones *) A1 = P0'P1; A2 = 1-2*P0^2; A3 = Q0'Q1; A4 = 1-2*Q0'2; A5 = r10'r13; A4 = 1-2*20'2; A7 = r20'r21; A8 = 1-2*r10'2; A7 = r30'r31; A10 = 1-2*r30'2; B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8; B3 = A10*A8 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6 A5R = R10*R13; A6R = 1-2*R10^2; A7R = R20*R21; A8R = 1-2*R20^2; A9R = R30*R31; A10R = 1-2*R30^2; B1R = A2*A8 + 4*A1*A7R; B2R = A1*A8R + 4*A1*A7R; B2R = A1*A8R + 4*A1*A7R; B3R = A10R*A7R - A8*A78; B4R = A10R*A7R - A8*A78; B4R = A2*A4 + 4*B4*B5R; C2R = B3R*B6R + 4*B4*B5R		
$\begin{array}{c} 2^{+}ARR^{+}(B2R^{+}L_{2}+C1R^{+}L_{3});\\ L1 - B1R^{+}L_{2}+C2R^{+}L_{3});\\ (* Componente 1 de la Base local del eslabón '(para dibujar la base) *)\\ e11R = {-A6R, 2^{+}A5R, 0};\\ (* Constantes de las ecuaciones *)\\ A1 = P0'P1; A2 = 1-2^{*}P0^{+}2;\\ A3 = Q0'Q1; A4 = 1-2^{*}Q0^{+}2;\\ A3 = Q0'Q1; A4 = 1-2^{*}Q0^{+}2;\\ A5 = r10^{*}r13; A6 = 1-2^{*}r10^{+}2;\\ A7 = r20^{*}r21; A8 = 1-2^{*}r20^{+}2;\\ A9 = r30^{*}r31; A10 = 1-2^{*}r30^{+}2;\\ B1 = A2^{+}A8 - 4^{+}A1^{+}A7; B2 = A2^{+}A7 + A1^{+}A8;\\ B3 = A10^{+}A8 - 4^{+}A1^{+}A7; B2 = A2^{+}A7 + A1^{+}A8;\\ B3 = A10^{+}A8 - 4^{+}A1^{+}A7; B2 = A2^{+}A7 + A1^{+}A8;\\ B5 = A2^{+}A4 - 4^{+}A1^{+}A3; B6 = A2^{+}A3 + A1^{+}A4;\\ C1 = B4^{*}B5 + B3^{*}B6; C2 = B3^{*}B5 - 4^{*}B4^{*}B6;\\ A5R = R10^{*}R13; A6R = 1-2^{*}R10^{+}2;\\ A9R = R30^{*}R31; A10R = 1-2^{*}R30^{+}2;\\ B1R = A2^{*}A8R + 4^{*}A1^{*}A7R;\\ B2R = A10^{*}R48 + 4^{*}A1^{*}A7R;\\ B2R = A10^{*}R47; A8R^{+}A9R;\\ B4R = A10R^{*}A7R - A2^{*}A7R;\\ B3R = A10R^{*}A7R + 4^{*}A1^{*}A3;\\ C1R = B4^{*}B6R - A2^{*}A3 + A1^{*}A4;\\ B6R = A2^{*}A4 - 4^{*}A1^{*}A3;\\ C1R = B4^{*}B6R + 4^{*}B4R^{*}B5R;\\ C2R = B3R^{*}B6R + 4^{*}B4R^{*}B5R;\\ C2R = B3R^{*}B6R + 4^{*}B4R^{*}B5R;\\ (* Longitud de los eslabones del manipulador *)\\ L1 = 6; L2 = 5; L3 = 4;\\ (* Respaldo de ecuaciones *)\\ z1 = b1; z2 = b2; z3 = b3; rx=p;\\ \end{array}$		$rpR = {4*A5R*(B2R*L2 + C1R*L3),$
L1 - B1R*L2 + C2R*L3); (* Componente 1 de la Base local del eslabón * (para dibujar la base) *) e11R = {-A6R, 2*A5R, 0}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1-2*P0^2; A3 = Q0*Q1; A4 = 1-2*Q0^2; A5 = r10*r13; A6 = 1-2*r10^2; A7 = r20*r21; A8 = 1-2*r30^2; B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8; B3 = A10*A8 - 4*A1*A3; B6 = A2*A3 + A1*A4; B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6 A5R = R10*R13; A6R = 1-2*R10^2; A7R = R20*R21; A8R = 1-2*R20^2; A9R = R30*R31; A10R = 1-2*R30^2; B1R = A2*A4 - 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A1*A7R; B3R = A10R*A8R + 4*A1*A7R; B3R = A10R*A8R + 4*A1*A7R; C1R = B4*B6R - B3R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaido de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=p; z1=b1; z2=b2; z3=b3; rx=p;		2*A6R*(B2R*L2 + C1R*L3),
(* Componente 1 de la Base local del eslabón ' (para dibujar la base) *) e11R = (-A6R, 2*A5R, 0); (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1-2*P0^2; A3 = Q0*Q1; A4 = 1-2*Q0^2; A5 = r10*r13; A6 = 1-2*r10*2; A7 = r20*r21; A8 = 1-2*r20*2; B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8; B3 = A10*A8 - 4*A7*A9; B4 = A10*A7 + A8*A6; B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6; A5R = R10*R13; A6R = 1-2*R20*2; A9R = R30*R31; A10R = 1-2*R20*2; A9R = R30*R31; A10R = 1-2*R20*2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A7R*A9R; B4R = A10R*A7R - A8R*A9R; B5R = A2*A4 - 4*A1*A3; C1R = B4*B5R - B3R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp; z1=b1; z2=b2; z3=b3; rx=rp;		L1 - B1R*L2 + C2R*L3};
(para dibujar la base) *) e11R = {-A6R, 2*A5R, 0}; (* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1-2*P0^2; A3 = Q0*Q1; A4 = 1-2*00*2; A5 = r10*r13; A6 = 1-2*r10*2; A7 = r20*r21; A8 = 1-2*r20*2; B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8; B3 = A10*A8 - 4*A7*A9; B4 = A10*A7 + A8*A6 B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6 A5R = R10*R13; A6R = 1-2*R20*2; A9R = R30*R31; A10R = 1-2*R30*2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A7R*A9R; B4R = A10R*A8R + 4*A7R*A9R; B4R = A10R*A8R + 4*A7R*A9R; B5R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp; z1=b1; z2=b2; z3=b3; rx=rp;		(* Componente 1 de la Base local del eslabón 1
$e11R = \{-A6R, 2^*A5R, 0\};$ (* Constantes de las ecuaciones *) A1 = P0'P1; A2 = 1-2*P0'2; A3 = Q0'Q1; A4 = 1-2*Q0'2; A5 = r10'r13; A6 = 1-2*r10'2; A7 = r20'r21; A8 = 1-2*r20'2; A9 = r30'r31; A10 = 1-2*r30'2; B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8; B3 = A10*R13; A6R = 1-2*R10'A7 + A8*A6; B5 = A2'A4 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6; A5R = R10*R13; A6R = 1-2*R10'A2; A7R = R20*R21; A8R = 1-2*R20'2; A9R = R30*R31; A10R = 1-2*R30'A2; B1R = A2*A8R + 4*A1*A7R; B2R = A10*R48 + 4*A1*A7R; B2R = A10*R48 + 4*A7R*A9R; B4R = A10R*A8R + A2*A7R; B3R = A10R*A8R + A2*A7R; B3R = A10R*A8R + A2*A7R; C1 = B4R*B6R - B3R*B6R; C1 = B4R*B6R + B3R*B5R; C2 = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) 21=b1; z2=b2; z3=b3; rx=rp;		(para dibujar la base) *)
(* Constantes de las ecuaciones *) A1 = P0*P1; A2 = 1-2*P0^2; A3 = Q0*Q1; A4 = 1-2*Q0^2; A5 = r10*r13; A6 = 1-2*r20^2; A7 = r20*r21; A8 = 1-2*r20^2; A9 = r30*r31; A10 = 1-2*r30^2; B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8; B3 = A10*A8 - 4*A7*A9; B4 = A10*A7 + A8*A6 B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6 A5R = R10*R13; A6R = 1-2*R10^2; A7R = R20*R21; A8R = 1-2*R20^2; A9R = R30*R31; A10R = 1-2*R30^2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A7R*A9R; B4R = A10R*A7R - A8R*A9R; B5R = A2*A3 + A1*A4; B6R = A2*A3 + A1*A4; B6R = A2*A3 + A1*A4; B6R = A2*A3 + A1*A4; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		e11R = {-A6R, 2*A5R, 0};
A1 = P0°P1; A2 = 1-2*P0^2; A3 = Q0°Q1; A4 = 1-2*Q0^2; A5 = r10*r13; A6 = 1-2*r10^2; A7 = r20*r21; A8 = 1-2*r30^2; B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8; B3 = A10*A8 - 4*A1*A3; B4 = A10*A7 + A8*A6 B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6 A5R = R10*R13; A6R = 1-2*R10^2; A7R = R20*R21; A8R = 1-2*R20^2; A9R = R30*R31; A10R = 1-2*R30^2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A1*A7R; B3R = A10R*A8R + 4*A1*A7R; B4R = A2*A8R + 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los esiabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaido de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp; z1=b1; z2=b2; z3=b3; rx=rp;		(* Constantes de las ecuaciones *)
A3 = Q0°Q1; A4 = $1-2^{*}Q0^{2}$; A5 = $r10^{*}r13$; A6 = $1-2^{*}r10^{2}$; A7 = $r20^{*}r21$; A8 = $1-2^{*}r20^{2}$; A9 = $r30^{*}r31$; A10 = $1-2^{*}r30^{2}$; B1 = A2^*A8 - $4^{*}A1^{*}A7$; B2 = $A2^{*}A7 + A1^{*}A8$; B3 = A10^*A8 - $4^{*}A1^{*}A3$; B6 = $A2^{*}A3 + A1^{*}A4$; C1 = B4*B5 + B3*B6; C2 = B3*B5 - $4^{*}B4^{*}B6$ A5R = R10^{*}R13; A6R = $1-2^{*}R10^{*}2$; A7R = R20^{*}R21; A8R = $1-2^{*}R20^{*}2$; A9R = R30^{*}R31; A10R = $1-2^{*}R30^{*}2$; B1R = $A2^{*}A8R + 4^{*}A1^{*}A7R$; B2R = $A1^{*}A8R - A2^{*}A7R$; B3R = $A10R^{*}A8R + 4^{*}A1^{*}A7R^{*}9R$; B3R = $A10R^{*}A8R + 4^{*}A1^{*}A7R^{*}9R$; B4R = $A10R^{*}A8R + 4^{*}A1^{*}A7R^{*}9R$; B4R = $A10R^{*}A8R + 4^{*}A1^{*}A7R^{*}9R$; B5R = $A2^{*}A4 - 4^{*}A1^{*}A3$; C1R = $B4R^{*}B6R - B3R^{*}B5R$; C2R = B3R^{*}B6R + 4^{*}B4R^{*}B5R; C2R = B3R^{*}B6R + 4^{*}B4R^{*}B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		A1 = P0*P1; A2 = 1-2*P0^2;
A5 = r10*r13; A6 = 1-2*r10*2; A7 = r20*r21; A8 = 1-2*r20*2; A9 = r30*r31; A10 = 1-2*r30*2; B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8; B3 = A10*A8 - 4*A7*A9; B4 = A10*A7 + A8*A6 B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6 A5R = R10*R13; A6R = 1-2*R10*2; A7R = R20*R21; A8R = 1-2*R20*2; A9R = R30*R31; A10R = 1-2*R30*2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A1*A7R; B3R = A10R*A7R - A8R*A9R; B4R = A10R*A7R - A8R*A9R; B5R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp; = C1R + 27 + 27 + 27 + 27 + 27 + 27 + 27 + 2		A3 = Q0°Q1; A4 = 1-2°Q0^2;
A7 = r20°r21; A8 = 1-2°r20°2; A9 = r30°r31; A10 = 1-2°r30°2; B1 = A2°A8 - 4°A1°A7; B2 = A2°A7 + A1°A8; B3 = A10°A8 - 4°A1°A3; B6 = A2°A3 + A1°A4; C1 = B4°B5 + B3°B6; C2 = B3°B5 - 4°B4°B6 A5R = R10°R13; A6R = 1-2°R10°2; A7R = R20°R21; A8R = 1-2°R20°2; A9R = R30°R31; A10R = 1-2°R30°2; B1R = A2°A8R + 4°A1°A7R; B2R = A1°A8R - A2°A7R; B3R = A10R°A8R + 4°A1°A7R; B4R = A10R°A7R - A8R°A9R; B5R = A2°A3 + A1°A4; B6R = A2°A3 + A1°A4; B6R = A2°A4 - 4°A1°A3; C1R = B4R°B6R - B3R°B5R; C2R = B3R°B6R + 4°B4R°B5R; (° Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (° Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		A5 = r10*r13; A6 = 1-2*r10^2;
A9 = r30°r31; A10 = 1-2°r30^2; B1 = A2°A8 - 4°A1°A7; B2 = A2°A7 + A1°A8; B3 = A10°A8 - 4°A1°A3; B4 = A10°A7 + A8°A8; B5 = A2°A4 - 4°A1°A3; B6 = A2°A3 + A1°A4; C1 = B4°B5 + B3°B6; C2 = B3°B5 - 4°B4°B6; A5R = R10°R13; A6R = 1-2°R10^2; A7R = R20°R21; A8R = 1-2°R20^2; A9R = R30°R31; A10R = 1-2°R30^2; B1R = A2°A8R + 4°A1°A7R; B2R = A1°A8R - A2°A7R; B3R = A10R°A7R - A8R°A9R; B4R = A10R°A7R - A8R°A9R; B5R = A2°A3 + A1°A4; B6R = A2°A3 + A1°A4; B6R = A2°A4 - 4°A1°A3; C1R = B4R°B6R - B3R°B5R; C2R = B3R°B6R + 4°B4R°B5R; (° Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (° Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		A7 = r20°r21; A8 = 1-2°r20^2;
B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8; B3 = A10*A8 - 4*A1*A3; B4 = A10*A7 + A8*A8; B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6; A5R = R10*R13; A6R = 1-2*R10^2; A7R = R20*R21; A8R = 1-2*R20^2; A9R = R30*R31; A10R = 1-2*R30^2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A1*A7R*A9R; B4R = A10R*A7R - A8R*A9R; B5R = A2*A3 + A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		A9 = r30*r31: A10 = 1-2*r30^2:
B3 = A10*A8 - 4*A7*A9; B4 = A10*A7 + A8*A6 B5 = A2*A4 - 4*A1*A3; B6 = A2*A3 + A1*A4; C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6 A5R = R10*R13; A6R = 1-2*R10^2; A7R = R20*R21; A8R = 1-2*R30^2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A7R*A9R; B4R = A10R*A7R - A8R*A9R; B5R = A2*A3 + A1*A4; B6R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		B1 = A2*A8 - 4*A1*A7; B2 = A2*A7 + A1*A8
$B_{5} = A_{2}^{*}A_{4} - 4^{*}A_{1}^{*}A_{3}; B_{6} = A_{2}^{*}A_{3} + A_{1}^{*}A_{4};$ $C_{1} = B_{4}^{*}B_{5} + B_{3}^{*}B_{6}; C_{2} = B_{3}^{*}B_{5} - 4^{*}B_{4}^{*}B_{6}$ $A_{5}R = R_{10}^{*}R_{13}; A_{6}R = 1-2^{*}R_{10}A_{2};$ $A_{7}R = R_{20}^{*}R_{21}; A_{8}R = 1-2^{*}R_{30}A_{2};$ $B_{1}R = A_{2}^{*}A_{8}R + 4^{*}A_{1}^{*}A_{7}R;$ $B_{2}R = A_{1}^{*}A_{8}R - A_{2}^{*}A_{7}R;$ $B_{3}R = A_{1}0R^{*}A_{8}R + 4^{*}A_{7}R^{*}A_{9}R;$ $B_{4}R = A_{1}0R^{*}A_{7}R - A_{8}R^{*}A_{9}R;$ $B_{5}R = A_{2}^{*}A_{3} + A_{1}^{*}A_{4};$ $B_{6}R = A_{2}^{*}A_{4} - 4^{*}A_{1}^{*}A_{3};$ $C_{1}R = B_{4}R^{*}B_{6}R - B_{3}R^{*}B_{5}R;$ $C_{2}R = B_{3}R^{*}B_{6}R + 4^{*}B_{4}R^{*}B_{5}R;$ $(* Longitud de los eslabones del manipulador *)$ $L_{1} = 6; L_{2} = 5; L_{3} = 4;$ $(* Respaldo de ecuaciones *)$ $z_{1}=b_{1}; z_{2}=b_{2}; z_{3}=b_{3}; nx=rp;$		B3 = A10*A8 - 4*A7*A9 B4 = A10*A7 + A8*A0
C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6 A5R = R10*R13; A6R = 1-2*R10^2; A7R = R20*R21; A8R = 1-2*R30^2; A9R = R30*R31; A10R = 1-2*R30^2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A7R*A9R; B4R = A10R*A8R + 4*A7R*A9R; B5R = A2*A3 + A1*A4; B6R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		B5 = A2*A4 - 4*A1*A3' B6 = A2*A3 + A1*A4
A5R = R10*R13; A6R = 1-2*R10^2; A7R = R20*R21; A8R = 1-2*R20^2; A9R = R30*R31; A10R = 1-2*R30^2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A7R*A9R; B4R = A10R*A7R - A8R*A9R; B5R = A2*A3 + A1*A4; B6R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		C1 = B4*B5 + B3*B6; C2 = B3*B5 - 4*B4*B6
A7R = R20*R21; A8R = 1-2*R20^2; A9R = R30*R31; A10R = 1-2*R30^2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A7R*A9R; B4R = A10R*A7R - A8R*A9R; B5R = A2*A3 + A1*A4; B6R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		A5R = R10*R13: A6R = 1-2*R10^2
A9R = R30*R31; A10R = 1-2*R30^2; B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A7R*A9R; B4R = A10R*A7R - A8R*A9R; B5R = A2*A3 + A1*A4; B6R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp; c1D= 1422 = 22=b22 = 22=b3; rx=rp; c1D= 1422 = 22=b		A7R = R20*R21' A8R = 1-2*R20^2
B1R = A2*A8R + 4*A1*A7R; B2R = A1*A8R - A2*A7R; B3R = A10R*A8R + 4*A7R*A9R; B4R = A10R*A7R - A8R*A9R; B5R = A2*A3 + A1*A4; B6R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los estabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		A9R = R30*R31' A10R = 1-2*R30^2
$B2R = A1^{*}A8R - A2^{*}A7R;$ $B3R = A10R^{*}A8R + 4^{*}A7R^{*}A9R;$ $B4R = A10R^{*}A7R - A8R^{*}A9R;$ $B5R = A2^{*}A3 + A1^{*}A4;$ $B6R = A2^{*}A4 - 4^{*}A1^{*}A3;$ $C1R = B4R^{*}B6R - B3R^{*}B5R;$ $C2R = B3R^{*}B6R + 4^{*}B4R^{*}B5R;$ $(* Longitud de los estabones del manipulador *)$ $L1 = 6; L2 = 5; L3 = 4;$ $(* Respaldo de ecuaciones *)$ $z1=b1; z2=b2; z3=b3; rx=rp;$		$B1R = A2^*A8R + A^*A1^*A7R$
B3R = A10R*A8R + 4*A7R*A9R; B3R = A10R*A7R - A8R*A9R; B4R = A10R*A7R - A8R*A9R; B5R = A2*A3 + A1*A4; B6R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		820 = A1*A90 - A2*A70
B3R = A10R A3R + 4 A/R A3R, B4R = A10R*A7R - A8R*A9R; B5R = A2*A3 + A1*A4; B6R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los estabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		220 = A100*A20 + A*A70*A00
B5R = A10K A/K - A0K A9K, B5R = A2*A3 + A1*A4; B6R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los estabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp; = 21=b1; z2=b2; z3=b3; rx=rp;		BAD - AIDRAD ADR T 4 A/K ASK,
B5R = A2*A3 + A1*A4; B6R = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp; z1=b1; z2=b2; z3=b3; rx=rp;		DER - ADTAS + ATTAA
<pre>BBR = A2*A4 - 4*A1*A3; C1R = B4R*B6R - B3R*B5R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los estabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp; c12=b1; c2=b2; c3=b3; rx=rp;</pre>		DOR - A2 A3 - A1 A4,
C1R = B4R*B6R - B3R*B6R; C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los estabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp; = 12=b1; z2=b2; z3=b3; rx=rp;		BOR = A2 A4 - 4 A1 A3;
C2R = B3R*B6R + 4*B4R*B5R; (* Longitud de los eslabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp; c12=b12; z2=b2; z3=b3; rx=rp;		CIR = 84R*86R - 83R*85R;
(* Longitud de los estabones del manipulador *) L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		C2R = B3R'B6R + 4"B4R'B5R;
L1 = 6; L2 = 5; L3 = 4; (* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		(* Longitud de los eslabones del manipulador *)
(* Respaldo de ecuaciones *) z1=b1; z2=b2; z3=b3; rx=rp;		L1 = 6; L2 = 5; L3 = 4;
z1=b1; z2=b2; z3=b3; rx=rp;		(* Respaldo de ecuaciones *)
		z1=b1; z2=b2; z3=b3; rx=rp;
	En esta parte se presenta el cálculo de la cinemática inversa para diferentes	CINEMATICA INVERSA
En esta parte se presenta el cálculo de la cinemática inversa para diferentes CINEMATICA INVERSA	trayectorias del elemento terminal, tanto para	
En esta parte se presenta el cálculo de la cinemática inversa para diferentes CINEMATICA INVERSA trayectorias del elemento terminal, tanto para el medio con retroinera como para el	modelo con rotaciones y reflexiones.	trayectorias
En esta parte se presenta el cálculo de la cinemática inversa para diferentes trayectorias del elemento terminal, tanto para el modelo con rotaciones, como para el modelo con rotaciones y reflexiones. CINEMATICA INVERSA Solución de ecuaciones para diferentes trayectorias	En esta parte se determinan los parámetros	LINEAS RECTAS
En esta parte se presenta el cálculo de la cinemática inversa para diferentes trayectorias del elemento terminal, tanto para el modelo con rotaciones, como para el modelo con rotaciones y reflexiones. CINEMATICA INVERSA Solución de ecuaciones para diferentes modelo con rotaciones y reflexiones. Solución de ecuaciones para diferentes trayectorias En esta parte se determinan los parámetros LINEAS RECTAS	de rotación de los eslabones, para una	
n esta parte se presenta el cálculo de la inemática inversa para diferentes ayectorias del elemento terminal, tanto para l modelo con rotaciones, como para el iodelo con rotaciones y reflexiones. n esta parte se determinan los parámetros e rotación de los eslabones, para una	ucesión de puntos que pertenecen a una	(* Se inicializan variables *)
En esta parte se presenta el cálculo de la cinemática inversa para diferentes rayectorias del elemento terminal, tanto para el modelo con rotaciones, como para el modelo con rotaciones y reflexiones. En esta parte se determinan los parámetros terrotación de los eslabones, para una sucesión de puntos que pertenecen a una (* Se inicializan variables *)	ínea recta, la cual se define por sus dos	ClearAll[r10,r13,r20,r21,r30,r31,"solucion"];

puntos extremos(S0, S1). Para lo anterior, primeramente recuperamos las ecuaciones del manipulador, las cuales se respaldaron en otras variables, (deberá resolverse la cinemática inversa para determinar la configuración no deformada y ejecutar la celda donde están las ecuaciones optimizadas para la configuración no deformada). Para determinar la puntos sobre la trayectoria, utilizamos la forma paramétrica de una recta que pasa por dos puntos $p_1 = \{x_1, y_1, z_1\}$ y $p_2 = \{x_1, y_1, z_1\}$:	ClearAll[R10,R13,R20,R21,R30,R31]; (* Se recuperan las ecuaciones del manipulador *) b1=z1; b2=z2; b3=z3; rp=rx; r p1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]]; b1R=z1R; b2R=z2R; b3R=z3R; rpR=rxR; rp1R=rpR[[1]];rp2R=rpR[[2]];rp3R=rpR[[3]]; (* Variables de control *) t = 0; (* Para discretizar en varios puntos la trayectoria seleccionada *) i = 1; (* Para guardar la solución de cada punto de la trayectoria *) (* Valores iniciales para resolver el sistema de
	ecuaciones *)
$px = x_1 + (x_2 - x_1)t$ $py = y_1 + (y_2 - y_1)t$ $pz = z_1 + (z_2 - z_1)t$	ir10 = .7071; ir13 = .7071; ir20 = .3; ir21 = .9; ir30 = 5; ir31 = .8;
	(* Travectoria en línea recta dada por los puntos
Con lo anterior se realiza un ciclo utilizando	extremos *)
como variable de control del mismo, a la variable t, con lo que determinamos puntos sobre la travectoria a seguir y resolvemos el	S0 = {5,5,7}; S1 = {5,-5,7};
sistema de ecuaciones (5.3), para el modelo con rotaciones.	(* Se resuelve la cinemática inversa para el modelo con Rotaciones *)
	(* Se inicia un ciclo para resoluer el problema
$r_{p}x = -4^{-}A5^{-}(B2^{-}L_{2} + C1^{-}L_{3}),$	cinematico inverso en los puntos en que sera
r.v= - 2*A6*(B2*L, + C1*L.).	discretizada la trayectoria *)
$r_p z = L_1 + B 1^{-} L_2 + C 2^{-} L_3$	FOTIT=0, t<1, t+=.1,
$r1_0^2 + r1_3^2 = 1^2$	(* Condeterminen les exerdenedes del sunte
	(* Se determinan las coordenadas del punto
rZ ₀ - + rZ ₁ - = 1	correspondiente de la trayectoria")
$r_{3}^{2} + r_{3}^{2} = 1$	$px = SO[[1]]+(S1[[1]]-SO[[1]])^{-t};$
	py = S0[[2]]+(S1[[2]]-S0[[2]])⁼t;
	pz = S0[[3]]+(S1[[3]]-S0[[3]])*t;
Este sistema de ecuaciones se resuelve utilizando la función Findroot usando los valores iniciales mostrados. Una vez que se	(* Solución del problema cinemático inverso para el punto seleccionado *)
resueive el sistema de ecuaciones anterior,	solucion[i] = FindRoot[{rp1==px, rp2==py, rp3==pz,
se verifica si la precisión de la solución es la	r10^2+r13^2 == 1, r20^2+r21^2 == 1,
adecuada, si es así, se fijan como valores	r30^2+r31^2 == 1}, {r10,ir10},{r13,ir13},
iniciales, pars resolver la cinemática inversa	{r20,ir20},{r21,ir21},{r30,ir30}, {r31,ir31},
del siguiente punto, a la solución obtenida y	MaxIterations->50];
se continua el ciclo, si la solución está fuera	
de la precisión se envía un mensaje y se	(* Se verifica la precisión de la solución *)
restauran los valores iniciales continuando el	e = 10^-5; (* Precisión requerida *)
ciclo hasta terminar. Después se realiza la	If[Abs[px - (rp1/ solucion[i])] <e &&<="" th=""></e>
solución del sistema de ecuaciones (5.5), con	Abs[py - (rp2/.solucion[i])] <e &&<="" th=""></e>
lo que se obtienen los parámetros de rotación	Abs(pz - (rp3/.solucion[i])] <e.< th=""></e.<>

correspondientes a al modelo con rotaciones y reflexiones.	(* Si la solución está en la precisión fijada, se definen como valores iniciales, para resolver la cinemática inversa del siguiente punto de la trayectoria a los valores de la solución obtenida, si
$r_p x = 4^*A5^*(B2^*L_2 + C1^*L_3)$	no es así se restauran los valores iniciales utilizados *)
r _p y= 2*A6*(B2*L _z + C1*L ₃)	ir10=r10/ solucion(i): ir13=r13/ solucion(i):
$r_{p}z = L_{1} - B1^{*}L_{2} + C2^{*}L_{3}$	ir20=r20/.solucion[i]; ir21=r21/.solucion[i];
$R1_0^2 + R1_3^2 = 1$	ir30=r30/.solucion[i]; ir31=r31/.solucion[i],
$R2_0^2 + R2_1^2 = 1$	(* Se imprime un mensaje si la solución no está en
R3, ² + R3, ² = 1	la precisión definida *)
	Print["No se encontró solución para el punto: ", {px,py,pz}];
	(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
	ir10 = .7071; ir13 = .7071; ir20 = .3; ir21 = .9; ir30 = .5; ir31 = .8];
	(* Se resuelve la cinemática inversa para el modelo con Reflexiones *)
	(* Se inicia un ciclo para resolver el problema cinemático inverso en los puntos en que será discretizada la trayectoria *)
	i = 1; (* Se inicializa la variable de control i *)
	For[t=0, t<1, t+=.1,
	(* Se determinan las coordenadas del punto correspondiente de la trayectoria*) px = S0[[1]]+(S1[[1]]-S0[[1]])*t; py = S0[[2]]+(S1[[2]]-S0[[2]])*t; pz = S0[[3]]+(S1[[3]]-S0[[3]])*t;
	(* Solución del problema cinemático inverso para el punto seleccionado *)
	solucionR[i] = FindRoot[{rp1R==px, rp2R==py, rp3R==pz, R10^2+R13^2 == 1, R20^2+R21^2 == 1, R30^2+R31^2 == 1}, {R10,ir10},{R13,ir13},{R20,ir20}, {R21,ir21}, {R30,ir30},{R31,ir31}, MaxIterations->50];
	(* Se verifica la precisión de la solución *)

	If[Abs[px - (rp1R/.solucion[i])] <e &&<br="">Abs[py - (rp2R/.solucion[i])]<e &&<br="">Abs[pz - (rp3R/.solucion[i])]<e,< th=""></e,<></e></e>
	(* Si la solución está en la precisión fijada, se definen como valores iniciales, para resolver la cinemática inversa del siguiente punto de la trayectoria a los valores de la solución obtenida, si no es así se restauran los valores iniciales utilizados *)
	ir10=R10/.solucion[i]; ir13=R13/.solucion[i]; ir20=R20/.solucion[i]; ir21=R21/.solucion[i]; ir30=R30/.solucion[i]; ir31=R31/.solucion[i];
	(* Se imprime un mensaje si la solución no está en la precisión definida *)
	Print["No se encontró solución para el punto: ", {px,py,pz}];
	(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
	ir10 = .7071; ir13 = .7071; ir20 = .3; ir21 = .9; ir30 = .5: ir31 = .8 1 _ i++ 1
En esta sección, al igual que en la anterior se	CIRCUNFERENCIAS O ELIPSES
En esta sección, al igual que en la anterior se resuelve la cinemática inversa para una	
En esta sección, al igual que en la anterior se resuelve la cinemática inversa para una secuencia de puntos sobre una trayectoria, para los dos modelos disponibles, que en este caso se trata de una elipse o circunferencia. El proceso que se sigue es el	CIRCUNFERENCIAS O ELIPSES (* Se inicializan variables *) ClearAll[r10,r13,r20,r21,r30,r31,"solucion"]; ClearAll[R10,R13,R20,R21,R30,R31];
En esta sección, al igual que en la anterior se resuelve la cinemática inversa para una secuencia de puntos sobre una trayectoria, para los dos modelos disponibles, que en este caso se trata de una elipse o circunferencia. El proceso que se sigue es el mismo que para la línea recta, solamente cambia la manera de determinar los puntos sobre la trayectoria, que en este caso se	CIRCUNFERENCIAS O ELIPSES (* Se inicializan variables *) ClearAll[r10,r13,r20,r21,r30,r31,"solucion"]; ClearAll[R10,R13,R20,R21,R30,R31]; (* Se recuperan las ecuaciones del manipulador *) b1=z1; b2=z2; b3=z3; rp=rx; r p1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]];
En esta sección, al igual que en la anterior se resuelve la cinemática inversa para una secuencia de puntos sobre una trayectoria, para los dos modelos disponibles, que en este caso se trata de una elipse o circunferencia. El proceso que se sigue es el mismo que para la línea recta, solamente cambia la manera de determinar los puntos sobre la trayectoria, que en este caso se realiza utilizando la forma paramétrica de una elipse en el plano y-z, con centro en {h, k, i} con radios radio1 y radio2.	CIRCUNFERENCIAS O ELIPSES (* Se inicializan variables *) ClearAll[r10,r13,r20,r21,r30,r31,"solucion"]; ClearAll[R10,R13,R20,R21,R30,R31]; (* Se recuperan las ecuaciones del manipulador *) b1=z1; b2=z2; b3=z3; rp=rx; r p1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]]; b1R=z1R; b2R=z2R; b3R=z3R; rpR=rxR; rp1R=rpR[[1]];rp2R=rpR[[2]];rp3R=rpR[[3]];
En esta sección, al igual que en la anterior se resuelve la cinemática inversa para una secuencia de puntos sobre una trayectoria, para los dos modelos disponibles, que en este caso se trata de una elipse o circunferencia. El proceso que se sigue es el mismo que para la línea recta, solamente cambia la manera de determinar los puntos sobre la trayectoria, que en este caso se realiza utilizando la forma paramétrica de una elipse en el plano y-z, con centro en {h, k, i} con radios <i>radio1</i> y <i>radio2</i> . px = h; py = radio1 * Cos[t]+k;	CIRCUNFERENCIAS O ELIPSES (* Se inicializan variables *) ClearAll[r10,r13,r20,r21,r30,r31,"solucion"]; ClearAll[R10,R13,R20,R21,R30,R31]; (* Se recuperan las ecuaciones del manipulador *) b1=z1; b2=z2; b3=z3; rp=rx; r p1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]]; b1R=z1R; b2R=z2R; b3R=z3R; rpR=rxR; rp1R=rpR[[1]];rp2R=rpR[[2]];rp3R=rpR[[3]]; (* Variables de control *) t = 0; (* Para discretizar en varios puntos la trayectoria seleccionada *)
En esta sección, al igual que en la anterior se resuelve la cinemática inversa para una secuencia de puntos sobre una trayectoria, para los dos modelos disponibles, que en este caso se trata de una elipse o circunferencia. El proceso que se sigue es el mismo que para la línea recta, solamente cambia la manera de determinar los puntos sobre la trayectoria, que en este caso se realiza utilizando la forma paramétrica de una elipse en el plano y-z, con centro en {h, k, i} con radios <i>radio1</i> y <i>radio2</i> . px = h; py = radio1 ° Cos[t]+k; pz = radio2 ° Sin[t]+l;	CIRCUNFERENCIAS O ELIPSES (* Se inicializan variables *) ClearAll[r10,r13,r20,r21,r30,r31,"solucion"]; ClearAll[R10,R13,R20,R21,R30,R31]; (* Se recuperan las ecuaciones del manipulador *) b1=z1; b2=z2; b3=z3; rp=rx; r p1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]]; b1R=z1R; b2R=z2R; b3R=z3R; rpR=rxR; rp1R=rpR[[1]];rp2R=rpR[[2]];rp3R=rpR[[3]]; (* Variables de control *) t = 0; (* Para discretizar en varios puntos la trayectoria seleccionada *) i = 1; (* Para guardar la solución de cada punto de la trayectoria *)
En esta sección, al igual que en la anterior se resuelve la cinemática inversa para una secuencia de puntos sobre una trayectoria, para los dos modelos disponibles, que en este caso se trata de una elipse o circunferencia. El proceso que se sigue es el mismo que para la línea recta, solamente cambia la manera de determinar los puntos sobre la trayectoria, que en este caso se realiza utilizando la forma paramétrica de una elipse en el plano y-z, con centro en {h, k, i} con radios radio1 y radio2. px = h; py = radio1 * Cos[t]+k; pz = radio2 * Sin[t]+l;	CIRCUNFERENCIAS O ELIPSES (* Se inicializan variables *) ClearAll[r10,r13,r20,r21,r30,r31,"solucion"]; ClearAll[R10,R13,R20,R21,R30,R31]; (* Se recuperan las ecuaciones del manipulador *) b1=z1; b2=z2; b3=z3; rp=rx; r p1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]]; b1R=z1R; b2R=z2R; b3R=z3R; rpR=rxR; rp1R=rpR[[1]];rp2R=rpR[[2]];rp3R=rpR[[3]]; (* Variables de control *) t = 0; (* Para discretizar en varios puntos la trayectoria seleccionada *) i = 1; (* Para guardar la solución de cada punto de la trayectoria *) (* Valores iniciales para resolver el sistema de ecuaciones *)
En esta sección, al igual que en la anterior se resuelve la cinemática inversa para una secuencia de puntos sobre una trayectoria, para los dos modelos disponibles, que en este caso se trata de una elipse o circunferencia. El proceso que se sigue es el mismo que para la línea recta, solamente cambia la manera de determinar los puntos sobre la trayectoria, que en este caso se realiza utilizando la forma paramétrica de una elipse en el plano y-z, con centro en {h, k, i} con radios <i>radio1</i> y <i>radio2</i> . px = h; py = radio1 ° Cos[t]+k; pz = radio2 ° Sin[t]+l;	CIRCUNFERENCIAS O ELIPSES (* Se inicializan variables *) ClearAll[r10,r13,r20,r21,r30,r31,"solucion"]; ClearAll[R10,R13,R20,R21,R30,R31]; (* Se recuperan las ecuaciones del manipulador *) b1=z1; b2=z2; b3=z3; rp=rx; r p1=rp[[1]]; rp2=rp[[2]]; rp3=rp[[3]]; b1R=z1R; b2R=z2R; b3R=z3R; rpR=rxR; rp1R=rpR[[1]];rp2R=rpR[[2]];rp3R=rpR[[3]]; (* Variables de control *) t = 0; (* Para discretizar en varios puntos la trayectoria seleccionada *) i = 1; (* Para guardar la solución de cada punto de la trayectoria *) (* Valores iniciales para resolver el sistema de ecuaciones *) ir10 = .7071; ir13 = .7071; ir20 = .3; ir21 = .9; ir30 = 5; ir31 = .8;



ir10 = .7071; ir13 = .7071; ir20 = .3; ir21 = .9; ir30 = .5; ir31 = .8]; i++]
(* Se resuelve la cinemática inversa para el modelo con Reflexiones *)
(* Se inicia un ciclo para resolver el problema cinemático inverso en los puntos en que será discretizada la trayectoria *)
i = 1; (* Se inicializa la variable de control i *)
For[t=0, t<=2*N[Pi], t+=N[Pi]/9,
(* Se determinan las coordenadas del punto correspondiente de la trayectoria*) px = h; py = radio1*Cos[t] + k; pz = radio2*Sin[t] + l;
(* Solución del problema cinemático inverso para el punto seleccionado *)
solucionR[i] = FindRoot[{rp1R==px, rp2R==py, rp3R==pz, R10^2+R13^2 == 1, R20^2+R21^2 == 1, R30^2+R31^2 == 1}, {R10,ir10},{R13,ir13},{R20,ir20}, {R21,ir21}, {R30,ir30},{R31,ir31}, Maxiterations->50];
(* Se verifica la precisión de la solución *)
lf[Abs[px - (rp1R/.solucion[i])] <e &&<br="">Abs[py - (rp2R/.solucion[i])]<e &&<br="">Abs[pz - (rp3R/.solucion[i])]<e,< th=""></e,<></e></e>
(* Si la solución está en la precisión fijada, se definen como valores iniciales, para resolver la cinemática inversa del siguiente punto de la trayectoria a los valores de la solución obtenida, si no es así se restauran los valores iniciales utilizados *)
ir10=R10/.solucion[i]; ir13=R13/.solucion[i]; ir20=R20/.solucion[i]; ir21=R21/.solucion[i]; ir30=R30/.solucion[i]; ir31=R31/.solucion[i];
(* Se imprime un mensaje si la solución no está en la precisión definida *)
Print["No se encontró solución para el punto: ", {px,py,pz}];
(* Se restauran valores iniciales para resolver las


r30^2+r31^2 == 1}, {r10,ir10},{r13,ir13}, {r20,ir20},{r21,ir21},{r30,ir30}, {r31,ir31}, MaxIterations->50];
(* Se verifica la precisión de la solución *)
e ≍ 10^-5; (* Precisión requerida *)
lf[Abs[px - (rp1/.solucion[i])] <e &&<="" th=""></e>
Abs[py - (rp2/.solucion[i])] <e &&<br="">Abs[pz - (rp3/.solucion[i])]<e,< th=""></e,<></e>
(* Si la solución está en la precisión fijada, se definen como valores iniciales, para resolver la cinemática inversa del siguiente punto de la trayectoria a los valores de la solución obtenida, si no es así se restauran los valores iniciales utilizados *)
ir10=r10/.solucion[i]; ir13=r13/.solucion[i]; ir20=r20/.solucion[i]; ir21=r21/.solucion[i]; ir30=r30/.solucion[i]; ir31=r31/.solucion[i],
(* Se imprime un mensaje si la solución no está en la precisión definida *)
Print["No se encontró solución para el punto: ", {px,py,pz}];
(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
ir10 = .7071; ir13 = .7071; ir20 = .3; ir21 = .9; ir30 = .5; ir31 = .8];
(* Se resuelve la cinemática inversa para el modelo con Reflexionee *)
(* Se inicia un ciclo para resolver el problema cinemático inverso en los puntos en que será discretizada la trayectoria *)
i ≖ 1; (* Se inicializa la variable de control i *)
For[t=0, t<=2*N[Pi], t+=N[Pi]/11,
(* Se determinan las coordenadas del punto correspondiente de la trayectoria*)
py = radio*Cos[t]^3 + k; pz = radio*Sin[t]^3 + l;
(* Solución del problema cinemático inverso para el punto seleccionado *)

	solucionR[i] = FindRoot[{rp1R==px, rp2R==py, rp3R==pz, R10^2+R13^2 == 1, R20^2+R21^2 == 1, R30^2+R31^2 == 1}, {R10,ir10},{R13,ir13},{R20,ir20}, {R21,ir21}, {R30,ir30},{R31,ir31}, Maxiterations->50]; (* Se verifica la precisión de la solución *) If[Abs[px - (rp1R/.solucion[i])] <e &&<br="">Abs[py - (rp2R/.solucion[i])]<e &&<br="">Abs[pz - (rp3R/.solucion[i])]<e &&<br="">Abs[pz - (rp3R/.solucion[i])]<e, (* Si la solución está en la precisión fijada, se definen como valores iniciales, para resolver la cinemática inversa del siguiente punto de la trayectoria a los valores de la solución obtenida, si no es así se restauran los valores iniciales utilizados *)</e, </e></e></e>
	ir10=R10/.solucion[i]; ir13=R13/.solucion[i]; ir20=R20/.solucion[i]; ir21=R21/.solucion[i]; ir30=R30/.solucion[i]; ir31=R31/.solucion[i];, (* Se imprime un mensaje si la solución no está en la precisión definida *)
	Print["No se encontró solución para el punto: ", {px,py,pz}];
	(* Se restauran valores iniciales para resolver las ecuaciones si no se encontró solución para el punto en cuestión *)
	ir10 = .7071; ir13 = .7071; ir20 = .3; ir21 = .9; ir30 = .5; ir31 =.8] i++]
Una vez determinados los parámetros de rotación para la trayectoria seleccionada, ahora se generarán las gráficas que permitirán ver la simulación computacional de	Graficación para visualizar la simulación de la trayectoria seleccionada. CiearAll[r10, r13, r20, r21, r30, r31, R10, R13, R20,
la cinemática inversa del manipulador de tres	R21, R30, R31]
grados de libertad. Para poder realizario, recuperamos los valores de los parámetros	(* Se recuperan los parámetros de rotación
almacenados en la variable solución[j] y los	almacenados en la variable solucion[j], obtenidos
aimacenamos en variables separadas para cada par de parámetros, se cenera una	almacenan en variables para cada par de
gráfica que servirá como base para graficar	parámetros *)
el manipulador, la cual se almacena en las	Forli=1 i <i i++="" r13)="" solr10]="/r10" solucion(i):<="" th=""></i>
inicia un ciclo, que se repetirá tantas veces	solr2[j]={r20,r21}/.solucion[j];
como valores de los parámetros se hayan	solr3[j]={r30,r31}/.solucion[j];
calculado para la trayectoria seleccionada	solR1[j]={R10,R13}/.solucionR[j];

(variable "i"). En cada pasada del ciclo se	solR2[j]={R20,R21}/ solucionR[j];
asignan los valores correspondientes a los	solR3[j]={R30,R31}/.solucionR[j]];
parametros y con ellos se evaluan	
numéricamente los dos conjuntos de	g1=g2=SurfaceGraphics[{{0,0,0},{0,0,0},{0,0,0}},
ecuaciones del manipulador, y con ellas se	MeshRange->{{-(L2+L3),L2+L3},
obtienen los puntos necesarios para graficar	{-(L2+L3),L2+L3}},
el manipulador en las dos configuraciones	AxesLabel->{Eje X,Eje Y,Eje Z},
correspondientes a los dos modelos	DefaultFont->{"Negrita", 12},
matemáticos. Para poder mostrar el efecto de	PiotRange->{{-(L2+L3+,1),(L2+L3+,1)},
la aplicación de los dos modelos	{-(L2+L3+.1).(L2+L3+.1)).
mencionados en el proceso de evasión de	$\{(1, 1, (1, 2+1, 3), (1, 1+1, 2+1, 3+, 1)\}\}$
obstáculos se define el obstáculo que será	RovRation->/1 1 1 Aves->Truel
un cubo unitario unicado de manera aleatoria	
dentre del área de trabaja del manimulador	/* Se define electorismente la posición del obstáculo
Centro del area de trabajo del manipulador.	
Con el obstaculo definido, se debe	[⁻)
determinar primeramente si para el modelo	
con puras rotaciones existe colisión entre uno	puntoi = {Random[Integer, {1,4}],
de los eslabones del manipulador y el	Random[integer,{-3,3}],Random[integer,{7,10}]);
obstáculo, si no es el caso, se graficará el	273 A 1071 - 104
manipulador con dicho modelo, pero si existe	(* Se inicia un ciclo para generar las gráficas de
colisión, se verificará si se produce colisión	cada punto de la trayectoria *)
para el modelo con rotaciones y reflexiones.	
graficando el manipulador con este modelo v	Forfi=1 i <i i++<="" td=""></i>
enviando en su caso un mensaie si aún no	
se nudo evadir el obstáculo con este cambio	(* Se asigna el valor de los parámetros para el punto
Dara daterminar el avieta colición entre el	de la travectoria correspondiente almacenados en
pelabán 2 a al 2 y al abatégula utilizamas al	
esiabon z o el 3 y el obstaculo, utilizamos el	las variables soir (Uj-)
calculo de la distancia mas corta entre los	
puntos que representan las esquinas del	r10=soir1[j][[1]]; r13=soir1[j][[2]];
cubo obstaculo y las líneas que representan	r20=soir2[j][[1]]; r21=soir2[j][[2]];
los eslabones 2 y 3, consideramos que existe	r30=solr3[[][[1]]; r31=solr3[]][[2]];
colisión cuando la distancia mencionada sea	R10=solR1[j][[1]]; R13=solR1[j][[2]];
menor que la unidad. El cálculo de dicha	R20=solR2[j][[1]]; R21=solR2[j][[2]];
distancia lo hacemos mediante el siguiente	R30=solR3[j][[1]]; R31=solR3[j][[2]];
procedimiento: En la siguiente figura se	
muestra el punto P y una gráfica de una recta	(* Se asignan valores a las ecuaciones del
que pasa por A y B, el punto M es el pie de la	manipulador, según los valores de los
perpendicular de P a la recta que pasa por A	parámetros, utilizando el respaldo de
	ecuaciones *)
, .	
	hase = N(e11); b1 = N(z1); b2 = N(z2); b3 = N(z3);
	$\frac{1}{1000} = \frac{1}{1000} = 1$
	$[10^{-10}](N)$, $[10^{-10}](N)$, $[10^{10}](N)$, $[10^{10}]$
	baser = N[e11r]; b1r = N[z1r]; b2r = N[z2r];
	psk = w[zsk]; tbk = w[rxk];
	(* Longitud del vector que servirá de base del
	manipulador *)
	a = 3;
	(* Se definen los puntos en el espacio tridimensional
	para graficar el manipulador tanto para rotaciones
	como para reflexiones *)



```
(* Eslabón 3, modelo con rotaciones *)
AB = b3:
AP = punto-(b1+b2);
nAP = Sqrt[AP[[1]]^2+AP[[2]]^2+AP[[3]]^2];
nAB = Sqrt[AB[[1]]^2+AB[[2]]^2+AB[[3]]^2];
AM = (AP.AB)/nAB;
PM2 = Sqrt[nAP^2-AM^2];
If[Im[PM2]!=0, PM2=0];
(* Si la distancia calculada para cualquiera de los
  dos eslabones fue menor a 1, la variable colision
  se iguala a 1 y se verifica ahora las distancias a
  los eslabones 2 y 3 para el modelo con
  reflexiones *)
If[PM1<=1 || PM2<=1, colision=1;
(* Eslabón 2, modelo con reflexiones *)
AB = b2R;
AP = punto-b1R:
nAP = Sqrt[AP[[1]]^2+AP[[2]]^2+AP[[3]]^2];
nAB = Sqrt[AB[[1]]^2+AB[[2]]^2+AB[[3]]^2];
AM = (AP.AB)/nAB;
PM3 = Sqrt[nAP^2-AM^2];
If[Im[PM3]!=0, PM3=0];
(* Eslabón 3, modelo con reflexiones *)
AB = b3R:
AP = punto-(b1R+b2R);
nAP = Sqrt[AP[[1]]^2+AP[[2]]^2+AP[[3]]^2];
nAB = Sqrt[AB[[1]]^2+AB[[2]]^2+AB[[3]]^2];
AM = (AP.AB)/nAB;
PM4 = Sqrt[nAP^2-AM^2];
If[Im[PM4]!=0, PM4=0];
(* Si cualquiera de las distancias calculadas es
  menor a 1, se asigna a la variable colision el valor
  de 2*)
If[PM3<=1 || PM4<=1, colision=2]]; ];
(* Se agrega a la gráfica base (g2) el obstáculo y la
  posición del elemento terminal *)
g2 = {Graphics3D[{{Cuboid[puntoi,puntof]},
                 {PointSize[0.01],Point[rp]},
                 {PointSize[0.01],Point[rpR]}}],g2};
(* Se genera la gráfica para manipulador con
  Rotaciones, si no hay colisión con el obstáculo
```

para este modelo (colisión = 0) *)
If[colision==0,
g3 = Graphics3D[{
(* Base rotatoria *) {AbsoluteThickness[40],RGBColor[0,.5,0], Line[{punto2,punto3}]}, (* Eslabón 1 *) {AbsoluteThickness[25],RGBColor[0,1,0], Line[{punto1,punto4}]], (* Eslabón 2 *) {AbsoluteThickness[20],RGBColor[1,0,0], Line[{punto4,punto5}]], (* Eslabón 3 *) {AbsoluteThickness[15],RGBColor[0,0,1], }
Line[{punto5,punto6}]])];,
(* Se genera la gráfica para manipulador con Reflexiones, si hay colisión con el obstáculo para el modelo con Rotaciones (colisión > 0), y en su caso, si tampoco se pudo evadir el obstáculo con el modelo con Reflexiones, se envía un mensaje indicándolo (colision > 1) *)
g3 = Graphics3D[{
(* Mensaje a emitir si no se pudo evadir el obstáculo *)
lf[colision==2, {Text["No se pudo evadir el obstáculo", {0,0,15}]},{}].
(* Base rotatoria *) {AbsoluteThickness[40],RGBColor[0,.5,0], Line[{punto2R,punto3R}]], (* Eslabón 1 *) {AbsoluteThickness[25],RGBColor[0,1,0], Line[{punto1R,punto4R}]], (* Eslabón 2 *) {AbsoluteThickness[20],RGBColor[1,0,0], Line[{punto4R,punto5R}]],
(* Esiabón 3 *) {AbsoluteThickness[15],RGBColor[0,0,1], Line[{punto5R,punto6R}]]}}]];
(* Se envían a pantalla las gráficas g1, g2 y g3, combinadas *)
Show[g1,g3,g2,Axes->True];]

EVASION DE OBSTACULOS EN LA CINEMATICA INVERSA



EVASION DE OBSTACULOS EN LA CINEMATICA INVERSA



CONCLUSIONES

- En l proceso de realización del presente trabajo se obtuvieron las siguientes conclusiones:
- S La aplicación del Lenguaje de Cálculo Formal Mathematica para efectuar demostraciones formales simbólicas, permite disminuir drásticamente el tiempo utilizado y los errores que se producen al realizar dichas demostraciones manualmente.
- ⊗ Las secuencias de movimientos, en el proceso de modelación, no afecta la forma final de las ecuaciones resultantes, pero se presentan diferentes configuraciones intermedias en cada secuencia.
- Las secuencias de movimientos pueden facilitar o dificultar la obtención de las ecuaciones cinemáticas finales, ya que en unas secuencias se requiere actualizar ejes de rotación en los Quaterniones correspondientes, lo que provoca que las ecuaciones resultantes sean más complejas y deban ser simplificadas utilizando propiedades de las operaciones con Quaterniones, mientras que en otras secuencias de movimientos no se requiere actualizar ejes de rotación, lo que permite que las ecuaciones obtenidas ya tengan su forma final simplificada.
- S La secuencia de movimientos más eficiente para modelar la cinemática de cadenas abiertas con Quaterniones, es iniciando con el eslabón terminal y finalizando con el eslabón base.
- S La forma general de la rotación de la base local de un eslabón cualquiera, en la configuración deformada de una cadena abierta con juntas rotacionales, es:

$$\mathbf{e}^{n}_{j} \equiv \rho(\mathbf{r}_{1}, \mathbf{r}_{2}^{*}...^{*}\mathbf{r}_{n}, \mathbf{e}^{x}_{j})$$

Donde:

eⁿ; = Base local del eslabón n.

- F₁, F₂, F_n = Quaterniones que rotan los eslabones 1, 2, ... n. Siendo el eslabón base el número uno, y el eslabón n, el eslabón analizado.
- e^x; = Base local del eslabón n en la configuración no deformada.

n = Número de eslabón analizado.

La numeración de los eslabones se realiza empezando en el eslabón base, es decir en el que se soporta todo el mecanismo, y se continua secuencialmente hasta finalizar en el eslabón terminal.

- Para modelar una traslación (junta prismática) con el álgebra de Quaterniones, basta con ubicar una base local en la junta, de la misma forma que se hace para las juntas rotacionales, y utilizar la iongitud del eslabón correspondiente como una variable la cual nos permitirá definir la traslación deseada.
- La combinación de rotaciones y reflexiones nos permite obtener un modelo cinemático deferente al que se obtiene utilizando solamente rotaciones, la aplicación conjunta de estos dos modelos para un manipulador de tres grados de libertad, permite que el manipulador logre evadir, en determinados casos, un obstáculo que impide su movimiento cuando se realiza un proceso de cinemática inversa.

REFERENCIAS

- [1] Reyes Avila Luis, 1990. Quaternions: Une Representation Parametrique Systematique Des Rotaions Finies.
 Partie 1: Le Cadre Theorique.
 Rapport de Recherche, INRIA No. 1303 - Rocquencourt, France.
- [2] Reyes Avila Luis, 1990. Quaternions: Une Representation Parametrique Systematique Des Rotaions Finies.
 Partie 2: Quelques Aplications.
 Rapport de Recherche, INRIA No. 1454 - Rocquencourt, France.
- [3] Stephen Wolfram, 1992. Mathematica a System for Doing Mathematics by Computer. Addison-Weley Publishing Company, Inc.
- [4] Cameron Smith and Nacy Blachman, 1995. The Mathematica Graphics Guidebook. Addison-Wesley Publishing Company, Inc.
- [5] Mark W. Spong and M. Vidyasagar, 1989. Robot Dynamics and Control. John Wiley & Sons.
- [6] Ilan Vardi, 1991. Computational Recreations in Mathematica. Addison-Wesley Publishing Company, Inc.
- [7] Stephen Wolfram, 1992. Mathematica Reference Guide. Addison-Wesley Publishing Company, Inc..
- [8] Louis Leithold, 1973. El cálculo con geometría analítica. HARLA.

LENGUAJE DE CALCULO FORMAL MATHEMATICA®.

En este apéndice se presentará el uso y sintaxis de los comandos de Mathematica utilizados en el desarrollo del presente trabajo. La primera versión de Mathematica se generó en 1988, se dice ser un sistema para hacer matemáticas en computadora. Combina manipulación simbólica, numérica, gráficos sobresalientes, y un idioma de programación sofisticado. A causa de su versatilidad, Mathematica se ha establecido como el sistema de álgebra en computadora elegido por muchos usuarios de la computadora. De los aproximadamente 100,000 usuarios de Mathematica, 28% son ingenieros, 21% son científicos en computación, 20% son científicos físicos, 12% son científicos matemáticos, y el 12% restante son hombres de negocios y científicos sociales. Dos terceras partes de los usuarios están en la industria y el gobierno, y solamente el 8% son estudiantes, este último porcentaje va en aumento. De cualquier modo, debido a su especial naturaleza y sofisticación, los usuarios principiantes necesitan conocer la sintaxis especial requerida para hacer que Mathematica ejecute la instrucciones en la manera que se desea. El propósito de este apéndice es servir como una guía a usuarios principiantes de Mathematica, para que logren comprender cabalmente los programas presentados en este trabajo. A continuación se presentan algunas reglas básicas en la sintaxis general en Mathematica.

Cinco Reglas Básicas de la sintaxis en Mathematica:

- 1.- Los argumentos de las funciones son dados entre paréntesis cuadrados [.], paréntesis redondos (.) se usan para agrupar operaciones; vectores, matrices, y las listas son dadas entre llaves (.); los paréntesis cuadrados dobles [[.]] se usan para indexación de listas y tablas, los paréntesis con asterisco (* *comentario* *) sirven para especificar comentarios.
- 2.- Los nombres de las funciones propias de Mathematica tienen su primera letra mayúscula; si un nombre de función consta de dos o más palabras, la primera letra de cada palabra será mayúscula sin dejar espacios intermedios..
- 3.- La multiplicación es representada por un espacio o por +.
- 4.- Las potencias son denotadas por ^A.
- 5.- Si no se consigue respuesta u obtiene una contestación incorrecta, puede deberse a una entrada o ejecución incorrecta de un comando. En algunos casos la cantidad de memoria asignada a Mathematica puede causar una calda del sistema; como algunas personas, Mathematica no es perfecto y algunos errores pueden ocurrir.

Ahora se mostraran los comandos, funciones y operaciones utilizadas, presentando una breve descripción, su sintaxis, y en algunos casos, un ejemplo.

 -> (Reglas de remplazo).- Una regla de remplazo, es una relación entre dos expresiones que son equivalentes, y que en un momento dado, se desea utilizar para remplazar una por la otra. Las reglas de remplazo se definen como: expr1 -> expr2, que significa que la expr1 se remplazará por expr2. Muchas de las funciones propias de Mathematica utilizan opciones, las cuales deben ser expresadas como reglas de remplazo.

- II. (Remplazo de expresiones).- Cuando se requiere sustituir dentro de una expresión, cualquier variable, utilizamos al final de la expresión el símbolo II. seguido por la lista de reglas de remplazo. Por ejemplo, si se desea remplazar el término (a²+b²) en la expresión (a²+b²)x + (a²+b²)y, se realiza lo siguiente: (a²+b²)x + (a²+b²)y/I.{(a²+b²)->c}, con lo que se obtiene como resultado cx + cy.
- AddTo.- x += dx suma dx a x y regresa el nuevo valor de x.
- ClearAll[symb1,symb2, ...].- Limpia todos los valores, definiciones, atributos, mensajes y asociaciones con los símbolos symboli. ClearAll["form1", "form2", ...] limpia todos los símbolos cuyos nombres coinciden con cualquiera de las formi.
- Collect[expr, {x1,x2, ... }].- Junta los términos que contienen la variable x1, x2, ... elevada a la misma potencia de xi. Ejemplos:

Collect[x + n x + m, x] ---> m + (1 + n) x; Collect[Expand[$(1+x+y)^3$], x] ---> 1 + x + 3 y + 3 y + y + x (3 + 3 y) + x (3 + 6 y + 3 y).

• Definición de Funciones.- Mathematica posee una gran cantidad de funciones propias, para realizar muchas operaciones, pero es común que el usuario requiera definir sus propias funciones. Por lo anterior mostraremos mediante un ejemplo la definición de una función, que para mayor compresión simplemente construiremos la función para determinar el producto escalar en R³. Si tenemos dos vectores a = {a_x, a_y, a_z} y b = {b_x, b_y, b_z}, sabemos que su producto escalar será: a•b = a_x*b_x + a_y*b_y + a_x*b_z, esto se programará en una función que llamaremos *ProductoEscalarR3*, como se muestra a continuación.

ProductoEscalarR3[a_,b_]:= a[[1]]*b[[1]] + a[[2]]*b[[2]] + a[[3]]*b[[3]];

En este caso la función tiene dos argumentos, que como se observa, deben encerrarse entre paréntesis cuadrados, y además, deben tener a continuación del nombre del parámetro un guión como se muestra. El signo := se utiliza para asignar a las función, las operaciones que se realizaran con los parámetros, que en este caso, es un producto escalar entre dos vectores. Finalmente el punto y coma ";" sirve para separar dos sentencias, o también, para evitar que una operación genere una salida.

Ejemplo de la ejecución en Mathematica de la función construida:

In[1] :=

```
(* Se define la función *)
ProductoEscalarR3[a_,b_]:= a[[1]]*b[[1]] + a[[2]]*b[[2]] + a[[3]]*b[[3]];
(* Se asignan componentes arbitrarias a dos vectores cualesquiera *)
x = {2,3,1}; y = {1,2,5};
```

(* Se aplica el producto *)

J

ProductoEscalarR3[x,y]

13

En este trabajo se definieron las siguientes funciones:

- ConjugadoQ[p].- Obtiene el Conjugado del Quaternión p, siendo p = {a, b, c, d} el conjugado de p = {a, b, c, d}.
- O Cruz[a,b].- Realiza el producto cruz de los vectores a,b ∈ ℜ³.
- ProductoQ[p,q].- Ejecuta la multiplicación de dos Quaterniones p y q, según la definición dada en la ecuación (2.2).
- Reflexion[p,v].- Obtiene la reflexión del vector v ∈ ℜ³, con el Quaternión p como parámetro. Para realizarla, primeramente se transforma el vector v a forma de Quaternión aplicándole TvInv[v] y se aplica la definición de reflexión (2.27), finalmente se transforma el Quaternión resultante de la rotación a su forma vectorial con la aplicación de Tv[], con lo que obtenemos el vector v en su nueva posición. Se considera que la norma del Quaternión p es unitaria.
- ReflexionQ[p,q].- Realiza la reflexión del Quaternión q con el Quaternión p como parámetro, utilizando la definición de reflexión (2.27) y sabiendo que al rotar un Quaternión solamente cambia su parte vectorial, conservando constante su parte escalar. Se considera que la norma del Quaternión p es unitaria.
- ◊ Rotacion[p,v].- Realiza la rotación del vector v ∈ ℜ³, con el Quaternión p como parámetro de rotación. Para realizarla, primeramente se transforma el vector v a forma de Quaternión aplicándole TvInv[v] y se aplica la definición de rotación (2.13), finalmente se transforma el Quaternión resultante de la rotación a su forma vectorial con la aplicación de Tv[], con lo que obtenemos el vector v en su nueva posición. Se considera que la norma del Quaternión p es unitaria.
- RotacionQ[p,q].- Rota el Quaternión q con el Quaternión p como parámetro de rotación, utilizando la definición de rotación (2.13) y sabiendo que al rotar un Quaternión solamente cambia su parte vectorial, conservando constante su parte escalar. Se considera que la norma del Quaternión p es unitaria.
- ◊ Tv[q].- Transforma el Quaternión q en un vector de R³. Es decir, si se aplica Tv[] al Quaternión p = {p₀, p₁, p₂, p₃}, se obtiene la parte vectorial de p, o sea:

$p_v = \{ p_1, p_2, p_3 \} = Tv[p]$

- Vinv[q].- Transforma un vector de R³ en un Quaternión agregando un cero como primer elemento del Quaternión.
- TvQ[q].- Transformación del Quaternión q en un Quaternión vectorial. Es decir, reemplaza el primer elemento del Quaternión (parte escalar) por cero.

- Expand[expr].- Esta función ejecuta los productos y potencias enteras positivas, asl como, las sumas en expr, hasta que todos los términos son monomiales, agrupando finalmente términos similares. Expand[] solo trabaja sobre potencias enteras positivas. Por ejemplo, si aplicamos está función a la expresión (x + y) ^ 2 + 7 (3 + x) (x + y), obtenemos como resultado 21x + 8x² + 21y + 9xy + y².
- Factor[poly].- Factoriza un polinomio sobre los enteros. Factor trabaja solamente con enteros exactos o coeficientes racionales, no con números reales. Factor[expr, Trig -> True] trata las funciones trigonométricas como funciones racionales de exponenciales, y las factoriza. Ejemplo: Factor[Sin[2x], Trig -> True] ---> 2 Cos[x] Sin[x].
- FindRoot[{eqn1, eqn2, ... }, {x, x0}, {y, y0}, ...].- Realiza la búsqueda de una solución numérica para las ecuaciones simultáneas eqni. FindRoot regresa una lista reglas de remplazo para x, y, ... en la misma forma que se obtiene de Solve. Si solamente se especifica un valor de inicio de x, FindRoot busca una solución usando el método de Newton. Si se especifican dos valores iniciales, FindRoot usa una variante del método de la secante. Una de las opciones que se pueden especificar es, MaxIterations que sirve para indicar el número de iteraciones que deberán realizarse. Ejemplo:

```
In[1] :=
```

```
(* Se resuelve numéricamente el siguiente sistema de ecuaciones *)
ec1 = 6*x + 8*y^2 == 30; (* Ecuación 1 *)
```

```
ec2 = 15*x^2 - 5*y == 40; (* Ecuación 2 *)
```

(* Utilizando el método de Newton, se aplica la opción MaxIterations para indicar que deseamos que se realicen 50 iteraciones, si no se indica se realizan 15 iteraciones *)

```
FindRoot[{ec1,ec2},{x,1},{y,1},MaxIterations->50]
```

```
(* Utilizando el método de la secante *)
```

FindRoot[{ec1,ec2},{x,1,2},{y,1,2},MaxIterations->50]

Out[5] := {x -> 1.78447, y -> 1.55295}

Out[7] := {x -> 1.78447, y -> 1.55295}

• Flatten[//sf].- Elimina los anidamientos de la lista //st. Ejemplos:

Flatten[{a, {b, c}, {d}}] ---> {a, b, c, d}.

 For[start, test, incr, body].- (Ciclo For) Ejecuta start, después evalúa repetidamente body y incr hasta que la comparación test sea falsa.

 Graphics3D[primitives, options].- Representa una imagen gráfica en tres dimensiones. Graphics3D despliega su contenido usando Show. Las siguientes tipos de gráficos primitives se pueden usar.

Cuboide Cuboid [xmin, ymin, zmin], ...] Linea Line[/(x1,y1,z1), ...]] punto Point[x, y, z]Poligono Polygon[([x1, y1, z1], ...]] Texto Text[expr, (x, y, z)]

Algunos de los modificadores para los gráficos que se utilizan son:

AbsoluteThickness[d]	Espesor absoluto de la línea.
PointSize[r]	Tamaño del punto especificado.
RGBColor[r, g, b]	Color del gráfico
as siguientes opciones se t	pueden especificar.

Las siguientes opciones se pueden especificar

Opción	Defauit	Descripción
Axes	False	Si dibuja ejes.
AxesLabel	None	Etiquetas de los ejes
BoxRatios	Automatic	Relación entre los lados de la caja 3D. que rodea el gráfico
DefaultFont	SDefaultFont	Tipo de letra a utilizar por defecto
PlotRange	Automatic	Rango de los valores a graficar

Un ejemplo se muestra en la Función Show.

- If[condition, t, f].- Devuelve t si la condición evaluada (condition) es verdadera, y f si es falsa. If condition, t, f, u) regresa u si la condición evaluada no es ni verdadera ni falsa. If solamente evalúa el argumento determinado por el valor de la condición.
- Im[z].- Proporciona la parte imaginaria de el número complejo z. Im[expr] se deja sin evaluar si expr no es numérica.
- Increment[x] or x++.- Incrementa el valor de x en una unidad y regresa el valor de x antes del incremento.
- Insert[/ist, elem, n].- Inserta elem en la posición n en la lista /ist. Si n es negativo, la posición es contada desde el fin de la lista. Insert[list, elem, {i, j,...}] inserta elem en la posición {i, j, ...} en list. Ejemplos:

 $lnsert[{a, b, c}, x, 2] ---> {a, x, b, c}.$ Insert[{a, b, c}, x, {{-1}, {1}}] --> (x, a, b, c, x). Insert[{(a, b), (c, d)}, x, {2, 1}] --> {(a, b), (x, c, d)}.

List[e1,e2, ...] = {e1,e2, ...}.- Es una lista de elementos. Las listas son objetos muy generales que representan colecciones de expresiones. $\{a, b, c\}$ representa un vector de \Re^3 , $\{\{a, b\}, \{c, d\}\}$

representa una matriz 2x2. Las listas pueden ser utilizadas también para representar tensores, Para hacer referencia a un elemento de una lista se utilizan los paréntesis cuadrados, elemplos:

 $a=\{x, y, z\}, A[[n]]$ se refiere al elemento n, A[[1]]=x, A[[2]]=y, A[[3]]=z $b = \{\{x_1, y_1, z_1\}, \{x_2, y_2, z_2\}, \{x_3, y_2, z_2\}\}, b[[n,m]]$ se refiere al elemento m de la sublista n.

- PowerExpandlexprl.- Expande todas las potencias de los productos y potencias de expr. Por ejemplo, PowerExpand[Sqrt[x y]] ---> Sqrt[x]*Sqrt[y]. PowerExpand convierte (a*b)^c a a^c*b^c. PowerExpand también convierte (a^b)^c a a^(b c).
- Randomí I.- Genera un número real pseudoaleaotorio uniformemente distribuído en el rango de 0 a 1. Random[type, range] regresa un número pseudoaleatorio del tipo especificado en type . perteneciente al rango range. Los tipos posibles son: Integer, Real y Complex. El rango por defecto es de 0 a 1. Se puede especificar el rango como {min, max}; si se especifica un rango de max es equivalente a {0, max}.
- ShowIgraphics, options].- Despliega en pantalla gráficas en dos y tres dimensiones (graphics) usando las opciones especificadas en options. Show[g1,g2, ...] muestra varias gráficas combinadas. Show puede usarse con Graphics, Graphics3D, SurfaceGraphics, ContourGraphics, DensityGraphics and GraphicsArray. Las opciones especificadas en Show eliminan las incluidas en la expresión gráfica graphics. Ejemplo:

In[1]:=





- Simplify[expr].- Realiza una secuencia de transformaciones algebraicas sobre expr, y regresa la forma más simple encontrada.
- Solve[eqs, vars].- Solve[eqs, vars] intenta resolver una ecuación o conjunto de ecuaciones para las variables vars. Solve[eqs, vars, elims] intenta resolver las ecuaciones para vars eliminando las variables elims. Las ecuaciones son dadas de la forma pi == pd. Ecuaciones simultáneas pueden especificarse en forma de lista. Se puede especificar una sola variable o una lista de ellas. Solve[eqs] trata de resolver para todas las variables en eqs. Ejemplo: Solve[3 x+ 9== 0, x]. Solve[] da soluciones explicitas como reglas de la forma x-> sol. Cuando hay varias variables, se da como una lista de reglas la solución: { x-> sx, y-> sy }. Cuando hay varias soluciones, Solve[] da una lista de ellas. Solve[] trata principalmente con ecuaciones lineales y polinomiales. Ejemplo:

 SurfaceGraphics[array].- Es una representación gráfica de una superficie tridimensional, con alturas de cada punto sobre una rejilla especificado por los valores en array. SurfaceGraphics[array, shades] representa una superficie, cuyas partes son sombreadas de acuerdo a la lista array. SurfaceGraphics puede se desplegada usando Show. SurfaceGraphics utiliza las mismas opciones que Graphics3D, con las siguientes adiciones:

Opción	Default	Descripción
HiddenSurface Mesh MeshStyle	True True Automatic	Si elimina superficies ocultas Si dibuja una malla sobre la superficie Especifica el estilo de para la dibujar la malla
MeshRange	Automatic	Rango x.y donde se dibujará la superficie

Un ejemplo se muestra en la función Show.

 Switch[expr, form1, value1, form2, value2, ...].- Evalúa expr, luego la compara con cada una de las formi, evaluando y regresando el valuei correspondiente a la primera igualdad encontrada. Solamente el valuei correspondiente al primer formi que es igual a expr será evaluado.

APENDICE B

PROGRAMAS INCLUIDOS EN DISCO.

En este apéndice se muestra el contenido del disco flexible incluido en esta tesis, dando una breve descripción y la forma de uso de cada uno de los programas que contiene.

- 1.- cap_2.ma ⇒ Demostraciones Capítulo II (Antecedentes). En este programa se presentan todas las demostraciones, del capítulo II, donde se utilizó Mathematica. El programa esta dividido en celdas que contienen cada una de las demostraciones realizadas. Para su utilización, basta simplemente con ejecutar la celda que contenga la demostración que deseamos verificar.
- 2.- 3gr_cin.ma ⇒ Cinemática de un robot manipulador de 3 grados de libertad utilizando rotaciones. En este programa se presenta la modelación y simulación de la cinemática de un robot manipulador de tres grados de libertad. Este programa se divide en las siguientes celdas principales:
 - a).- Definición de la Transformación Rotación. En esta celda se define la transformación rotación y otras definiciones necesarias.
 - b).- Definiciones iniciales. Aquí se define la base canónica de R³ y las bases locales correspondientes a cada eslabón del manipulador, así como los Quaterniones necesarios para la modelación.
 - c).- MODELADO DE LA CONFIGURACION NO DEFORMADA. Ceida título.
 - d).- Modelo para la configuración no deformada (Secuencia 3-2). Se realiza la obtención de las ecuaciones cinemáticas correspondientes a la configuración no deformada.
 - e).- Ecuaciones optimizadas para la configuración no deformada. Se presentan las ecuaciones obtenidas en la celda anterior, con las cuales se determinarán los parámetros de rotación correspondientes a la configuración no deformada.
 - f).- Cinemática inversa para determinar la configuración no deformada. En esta celda se realiza el proceso de cinemática inversa para determinar la configuración no deformada correspondiente a una posición del elemento terminal del robot.
 - g).- MODELADO DE LA CONFIGURACION DEFORMADA Celda título.
 - h).- Modelo para la configuración deformada (Secuencia 3-2-1). Se obtienen las ecuaciones cenemáticas para la configuración deformada del robot de tres grados de libertad.
 - i).- Ecuaciones optimizadas para la configuración deformada. Se presentan las ecuaciones obtenidas en la celda anterior, las cuales serán utilizadas para poder realizar el proceso de simulación de la cinemática del robot.
 - j).- CINEMATICA DIRECTA. Celda título.
 - k).- Graficación para visualizar la simulación. Esta celda permite generar las gráficas necesarias para generar una simulación de la cinemática directa.
 - I).- CINEMATICA INVERSA. Ceida titulo.
 - m).-Solución de ecuaciones para diferentes trayectorias. Celda título.
 - n).- <u>LINEAS RECTAS.</u> En esta celda se determinan los parámetros de rotación necesarios para que el elemento terminal del robot siga una trayectoria en línea recta, la cual es discretizada en una serie de puntos.
 - o).- <u>CIRCUNFERENCIAS O ELIPSES.</u> En esta celda se determinan los parámetros de rotación necesarios para que el elemento terminal del robot siga una trayectoria en forma de elipse o circunferencia, la cual es discretizada en una serie de puntos.

- p).- <u>ASTROIDE.</u> En esta celda se determinan los parámetros de rotación necesarios para que el elemento terminal del robot siga una trayectoria en forma de astroide, la cual es discretizada en una serie de puntos.
- q).- Graficación para visualizar la simulación de la trayectoria seleccionada. Finalmente, esta celda nos genera las gráficas necesaria para observar la simulación de la cinemática inversa del robot en cuestión. La simulación puede ser de cualquiera de las trayectorias programadas en las celdas anteriores. la que se obtenga dependerá de cual fue la celda de trayectoria que se ejecutó anteriormente.

Si deseamos generar las gráficas necesarias para observar la simulación de la cinemática del robot, debemos seguir una secuencia de ejecución de celdas, es decir debemos ejecutar en secuencia las siguientes celdas: **a**, **d**, **e**, **g**, **i**, para la cinemática directa. Para la cinemática inversa, si ya se ejecutaron las anteriores, se debe ejecutar una de las celdas correspondiente a una trayectoria (**n**, **o**, **p**) y la celda **q**. Una vez generadas las gráficas, basta con seleccionar la celda que las contiene y presionar el icono correspondiente para observar la simulación.

- 3.- 4gr_cin.ma ⇒ Cinemática de un robot manipulador de 4 grados de libertad utilizando rotaciones. En este programa se presenta la modelación y simulación de la cinemática de un robot manipulador de cuatro grados de libertad, el cual se presentó en el capítulo III. Este programa se divide las mismas celdas que el anterior y se ejecuta de la misma forma.
- 4.- Scara.ma ⇒ Cinemática de un robot manipulador de 4 grados de libertad tipo SCARA utilizando rotaciones. En este programa se presenta la modelación y simulación de la cinemática de un robot manipulador tipo SCARA. Sus celdas y modo de ejecución son similares a los anteriores.
- 5.- Evasion.ma ⇒ Ejemplo de aplicación de la modelación con una combinación de rotaciones y reflexiones de un robot manipulador de 3 grados de libertad en el proceso de evasión de obstáculos. En este programa se presenta la modelación de la cinemática de un robot manipulador de tres grados de libertad utilizando una combinación de rotaciones y reflexiones, y se aplican para mostrar la simulación de la cinemática inversa con un proceso de evasión de obstáculos. Este programa se divide en la siguientes celdas:
 - a).- Definición de las Transformaciones Rotación y Reflexión. En esta ceida se definen las transformaciones rotación y reflexión, así como algunas otras definiciones necesarias.
 - b).- Definiciones iniciales. Aquí se define la base canónica de R³ y las bases locales correspondientes a cada estabón del manipulador, así como los Quaterniones necesarios para la modelación.
 - c).- MODELADO DE LA CONFIGURACION NO DEFORMADA. Ceida título.
 - d).- Modelo para la configuración no deformada. Se realiza la obtención de las ecuaciones cinemáticas correspondientes a la configuración no deformada.
 - e).- Ecuaciones optimizadas para la configuración no deformada. Se presentan las ecuaciones obtenidas en la celda anterior, con las cuales se determinarán los parámetros de rotación correspondientes a la configuración no deformada.
 - f).- Cinemática inversa para determinar la configuración no deformada. En esta celda se realiza el proceso de cinemática inversa para determinar la configuración no deformada correspondiente a una posición del elemento terminal del robot.
 - g).- MODELADO DE LA CONFIGURACION DEFORMADA Ceida título.
 - h).- Modelo para la configuración deformada con rotaciones. Se obtienen las ecuaciones cenemáticas para la configuración deformada del robot de tres grados de libertad, utilizando solamente rotaciones.
 - i).- Modelo para la configuración deformada con rotaciones y reflexiones. Se obtienen las ecuaciones cenemáticas para la configuración deformada del robot de tres grados de libertad, utilizando una combinación de rotaciones y reflexiones.

- j).- Ecuaciones optimizadas para la configuración deformada. Se presentan las ecuaciones obtenidas en la celdas anterior, las cuales serán utilizadas para poder realizar el proceso de simulación de la cinemática del robot.
- k).- CINEMATICA INVERSA. Celda título.
- I).- Solución de ecuaciones para diferentes trayectorias. Celda título.
- m).-<u>LINEAS RECTAS.</u> En esta celda se determinan los parámetros de rotación necesarios para que el elemento terminal del robot siga una trayectoria en línea recta, la cual es discretizada en una serie de puntos.
- n).- <u>CIRCUNFERENCIAS O ELIPSES.</u> En esta ceida se determinan los parámetros de rotación necesarios para que el elemento terminal del robot siga una trayectoria en forma de elipse o circunferencia, la cual es discretizada en una serie de puntos.
- o).- <u>ASTROIDE.</u> En esta celda se determinan los parámetros de rotación necesarios para que el elemento terminal del robot siga una trayectoria en forma de astroide, la cual es discretizada en una serie de puntos.
- p).- Graficación para visualizar la simulación de la trayectoria seleccionada. Finalmente, esta celda nos genera las gráficas necesaria para observar la simulación de la cinemática inversa del robot en cuestión, ubicando un obstáculo dentro del espacio de trabajo del robot de manera aleatoria y decidiendo que modelo cinemático utilizar para evadir dicho obstáculo. La simulación puede ser de cualquiera de las trayectorias programadas en las celdas anteriores, la que se obtenga dependerá de cual fue la celda de trayectoria que se ejecutó anteriormente.

Para lograr obtener las gráficas necesarias para observar la simulación deseada, se debe realizar la ejecución de las siguientes celdas: primeramente a, e, f, i, después la celda correspondiente a la trayectoria deseada (m, n, o), y finalmente la celda p con lo que se generarán las gráficas para la simulación correspondiente.