

**APLICACIÓN DE LA HIPÓTESIS *VIRTUAL SPRING DAMPER* EN EL
CONTROL DE POSICIÓN DE UN DEDO DE PRÓTESIS DE MANO
ROBÓTICA SUB-ACTUADA: ESTUDIO DE SIMULACIÓN**



Mónica Isabel Muñoz Muñoz

Juan Felipe Solarte Paredes

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Ingeniería en Automática Industrial
Popayán, 2017**

**APLICACIÓN DE LA HIPÓTESIS *VIRTUAL SPRING DAMPER* EN EL
CONTROL DE POSICIÓN DE UN DEDO DE PRÓTESIS DE MANO
ROBÓTICA SUB-ACTUADA: ESTUDIO DE SIMULACIÓN**



**Trabajo de grado presentado para optar al título de Ingenieros en
Automática Industrial**

Mónica Isabel Muñoz Muñoz

Juan Felipe Solarte Paredes

Director: PhD. Carlos Alberto Gaviria López

**Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Electrónica, Instrumentación y Control
Ingeniería en Automática Industrial
Popayán, 2017**

Agradecimientos

A nuestras familias y especialmente a nuestros padres y hermanos, por siempre brindarnos su apoyo constante y sincero, por ser pilares fundamentales en nuestra vida y darnos las fuerzas y ánimo necesarios para alcanzar nuestros sueños.

Gracias a nuestros compañeros y amigos que nos ofrecieron su valiosa amistad, por compartir con nosotros su tiempo, espacio, conocimientos, y alegrías.

De forma especial a nuestro director, PhD. Carlos Alberto Gaviria López, por los conocimientos compartidos a lo largo de este proyecto, por la paciencia, tiempo y dedicación para hacer de esto algo posible.

TABLA DE CONTENIDO

1.1	Cadena planar	1
1.2	Grados de libertad (GDL)	2
1.3	Modelo Geométrico	2
1.4	Restricciones cinemáticas	4
1.4.1	Ecuaciones de cierre	4
1.4.1	Solución de las ecuaciones de restricción	5
1.5	Modelo cinemático directo	7
1.5.1	Cinemática de sistemas con cadenas cinemáticas cerradas	7
1.5.2	Cinemática diferencial	9
1.6	Mecanismos sub-actuados	10
1.6.1	Diseño de un dedo antropomórfico sub-actuado	11
1.6.2	Análisis de posición para el dedo	11
1.6.3	Ecuaciones de restricción para el dedo sub-actuado	21
1.6.4	Solución de las ecuaciones de restricción	22
1.7	Análisis cinemático dedo sub-actuado	25
2	Representación del modelo dinámico usando Adams	28
2.1	Generación del modelo en Adams	28
3	HIPÓTESIS VIRTUAL SPRING DAMPER	33
3.1	Enfoque virtual spring damper para el modelado y simulación de sistemas robóticos con cadenas cinemáticas cerradas	33
3.2	Comparación entre el algoritmo de control virtual spring damper y las estrategias de control más utilizadas	34
3.2.1	Control PID.	35
3.2.1.1	PID Articular.	35
3.2.1.2	PID Cartesiano.	37
3.2.2	Control por par calculado (CTC).	37
3.2.2.1	CTC Articular.	37
3.2.2.2	CTC Operacional.	37
3.2.3	Algoritmo de control basado en la hipótesis virtual spring damper.	38
3.2.3.1	Agarre de una prótesis de mano utilizando el algoritmo de control virtual spring damper.	49

3.2.3.1.1	Precisión en el agarre	50
3.2.3.1.2	Poder en el agarre.....	50
4	OPTIMIZACIÓN PARA SINTONIZAR CONTROL VSD y PID	52
4.1	Optimización para sintonizar el control PID	54
4.2	Optimización para sintonizar el control VSD.....	59
5	RESULTADOS	63
6	CONCLUSIONES Y TRABAJOS FUTUROS	90

LISTA DE FIGURAS

Figura 1. Cadenas planares.....	1
Figura 2. Mecanismo de cuatro barras	4
Figura 3. CAD dedo índice sub-actuado	11
Figura 4. Mecanismo doble de 4 barras cruzadas	12
Figura 5. Mecanismo 4 barras cadena interna.....	12
Figura 6. Cadena externa	12
Figura 7. Falange Proximal.....	13
Figura 8. Falange Media	13
Figura 9. Falange Distal.....	14
Figura 10. Modelo del dedo con los marcos coordenados D-H.	15
Figura 11. Parámetros D-H cadena cinemática abierta 1.	16
Figura 12. Cadena cinemática abierta 1.	16
Figura 13. Parámetros D-H cadena cinemática abierta 2	17
Figura 14. Cadena cinemática abierta 2.	17
Figura 15. Parámetros D-H cadena cinemática abierta 3	18
Figura 16. Cadena cinemática abierta 3.	18
Figura 17. Parámetros D-H cadena cinemática abierta 4.	19
Figura 18. Cadena cinemática abierta 4.	19
Figura 19. Cinemática directa cadena abierta 1.....	20
Figura 20. Cinemática directa cadena abierta 2.....	20
Figura 21. Cinemática directa cadena abierta 3.....	20
Figura 22. Cinemática directa cadena abierta 4.....	21
Figura 23. Experimento en MATLAB validación de ángulos	26
Figura 24. Representación entrada-salida de Adams	28
Figura 25. Representación en lazo cerrado de un sistema usando Adams	28
Figura 26. Modelo Adams_sys.....	30
Figura 27. Bloque adams_sub	30
Figura 28. Modelo bajo la acción de la fuerza de gravedad sin objeto en el camino y con objeto en el camino.....	30
Figura 29. Vista renderizada de un ambiente simulado, con una mesa y un objeto	31
Figura 30. Simulink modelo del dedo en Adams.....	32
Figura 31. Jacobiana obtenida en simulación con Adams	32
Figura 32. Esquema general control PID (Tomado de [10]).....	35
Figura 33. Esquema general control PID (Tomado de [10]).....	39
Figura 34. Resorte y amortiguador en paralelo.....	41
Figura 35. Hipótesis virtual spring damper aplicada a un brazo robótico (Tomada de [12])	47
Figura 36. Esquema del control PID	54
Figura 37. Trayectoria articular deseada	55
Figura 38. Error de seguimiento de la trayectoria articular	55
Figura 39. Esfuerzo de control	55

Figura 40. Seguimiento de la trayectoria articular deseada	56
Figura 41. Trayectoria cartesiana obtenida.....	56
Figura 42. Error articular	57
Figura 43. Seguimiento de la trayectoria articular.....	58
Figura 44. Trayectoria cartesiana obtenida.....	58
Figura 45. Esquema del control VSD.....	59
Figura 46. Error en el alcance del punto deseado	60
Figura 47. Esfuerzo del controlador VSD.....	60
Figura 48. Trayectoria de la yema del dedo para alcanzar el punto deseado..	60
Figura 49. Errores cartesianos obtenidos	61
Figura 50. Alcance del punto deseado.....	62
Figura 51. Trayectoria deseada	63
Figura 52. Error articular	63
Figura 53. Esfuerzo de control.....	64
Figura 54. Seguimiento de trayectoria	64
Figura 55. Trayectoria de referencia en el espacio operacional.....	65
Figura 56. Error de seguimiento.....	65
Figura 57. Torque aplicado a la articulación activa	65
Figura 58. Seguimiento de la trayectoria deseada.....	65
Figura 59. Simulación en Adams	66
Figura 60. Error articular	67
Figura 61. Esfuerzo de control.....	67
Figura 62. Trayectoria obtenida	67
Figura 63. Error en el alcance del punto deseado	68
Figura 64. Esfuerzo del controlador VSD.....	68
Figura 65. Trayectoria de la yema del dedo para alcanzar el punto deseado...	69
Figura 66. Simulación en Adams.	69
Figura 67. Error en el alcance del punto deseado	70
Figura 68. Esfuerzo del controlador VSD.....	70
Figura 69. Trayectoria de la yema del dedo para alcanzar el punto deseado...	71
Figura 70. Simulación en Adams	71
Figura 71. Trayectoria deseada	72
Figura 72. Error articular	72
Figura 73. Esfuerzo de control.....	73
Figura 74. Trayectoria obtenida	73
Figura 75. Trayectoria de referencia en el espacio operacional.....	74
Figura 76. Error de seguimiento de la trayectoria deseada	74
Figura 77. Torque aplicado a la articulación activa	74
Figura 78. Seguimiento de la trayectoria deseada.....	74
Figura 79. Simulación en Adams	74
Figura 80. Error articular	76
Figura 81. Esfuerzo de control.....	76
Figura 82. Trayectoria obtenida	76
Figura 83. Error en el alcance del punto deseado	77
Figura 84. Esfuerzo del controlador VSD.....	77

<i>Figura 85. Trayectoria de la yema del dedo para alcanzar el punto deseado...</i>	<i>77</i>
<i>Figura 86. Simulación en Adams</i>	<i>77</i>
<i>Figura 87. Error en el alcance del punto deseado</i>	<i>79</i>
<i>Figura 88. Esfuerzo del controlador VSD.....</i>	<i>79</i>
<i>Figura 89. Trayectoria de la yema del dedo para alcanzar el punto deseado...</i>	<i>79</i>
<i>Figura 90. Simulación en Adams</i>	<i>79</i>
<i>Figura 91. Diagrama de bloques controlador PID cartesiano</i>	<i>83</i>
<i>Figura 92. Esfuerzo de control.....</i>	<i>84</i>
<i>Figura 93. Error cartesiano</i>	<i>84</i>
<i>Figura 94. Seguimiento de la referencia</i>	<i>84</i>
<i>Figura 95. Trayectoria del dedo en el plano xy.....</i>	<i>84</i>
<i>Figura 96. Esfuerzo de control.....</i>	<i>85</i>
<i>Figura 97. Error cartesiano</i>	<i>85</i>
<i>Figura 98. Seguimiento de la referencia</i>	<i>85</i>
<i>Figura 99. Trayectoria del dedo en el plano xy resaltando el punto inalcanzable escogido</i>	<i>85</i>
<i>Figura 100. Esfuerzo de control.....</i>	<i>86</i>
<i>Figura 101. Error cartesiano</i>	<i>86</i>
<i>Figura 102. Seguimiento de la referencia</i>	<i>87</i>
<i>Figura 103. Trayectoria del dedo en el plano xy.....</i>	<i>87</i>
<i>Figura 104. Esfuerzo de control.....</i>	<i>88</i>
<i>Figura 105. Error cartesiano</i>	<i>88</i>
<i>Figura 106. Seguimiento de la referencia</i>	<i>88</i>
<i>Figura 107. Trayectoria del dedo en el plano xy.....</i>	<i>88</i>

1. MODELO CINEMÁTICO PARA CADENAS CERRADAS

Un sistema de cuerpos rígidos conectados entre sí por juntas recibe el nombre de cadena cinemática. Una cadena cinemática se denomina cerrada, si esta forma una cadena de lazo cerrado en la cual, cada uno de los cuerpos que la componen puede tener múltiples predecesores y múltiples sucesores. Una cadena cinemática se denomina serial, si cada uno de los cuerpos que la compone tiene un solo predecesor y un solo sucesor.

Principalmente hay cuatro tipos de articulaciones que se pueden encontrar en estos mecanismos:

- Articulación rotoide o de revolución (R)
- Articulación prismática o deslizante (P)
- Articulación esférica o de bola (S)
- Articulación helicoidal o de tornillo (H)

1.1 Cadena planar

Una cadena planar, como se ilustra en la Figura 1, es una cadena cinemática en la cual todos los eslabones de la cadena están restringidos a moverse en paralelo al mismo plano [1]. Estas cadenas admiten únicamente juntas prismáticas y de revolución, de hecho, los ejes de las articulaciones rotoides deben ser perpendiculares al plano de la cadena, mientras que los ejes de las articulaciones prismáticas deben ser paralelos o estar situados en el plano de la cadena.

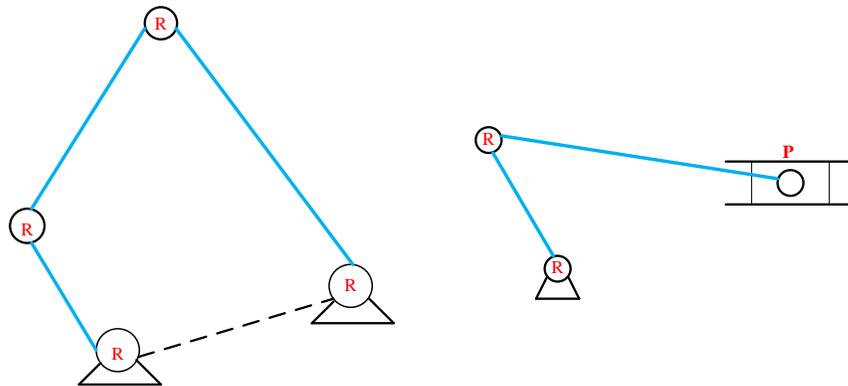


Figura 1. Cadenas planares

1.2 Grados de libertad (GDL)

El grado de movilidad de un mecanismo es el número de variables independientes requeridas para definir su posición o también el número de eslabones que hay que accionar para controlar el movimiento del mecanismo. El número de grados de libertad de un mecanismo con eslabones y juntas puede ser calculado usando la fórmula de Grübler [2] así:

$$GDL = m(N - 1 - J) + \sum_{i=1}^J f_i \quad (1)$$

Dónde:

m : 3 Para mecanismos planos y 6 para mecanismos espaciales.

N : Número de eslabones donde también se considera el de tierra como uno.

J : Numero de juntas.

f_i : Número de grados de libertad de la articulación.

1.3 Modelo Geométrico

El modelo geométrico expresa la situación del órgano terminal, es decir las coordenadas operacionales del robot, en función de sus coordenadas articulares. De otro modo, conocidas las posiciones articulares de cada articulación de un robot, este modelo permite conocer la posición cartesiana y la orientación del órgano terminal.

El algoritmo propuesto por Denavit-Hartenberg [3] (D-H), permite relacionar los marcos coordenados asociados a un eslabón con los de los marcos de los siguientes eslabones mediante 4 transformaciones básicas, las cuales dependen exclusivamente de las características constructivas del robot, dados los sistemas de referencia de todos los eslabones de las cadenas cinemáticas del mecanismo. Estas transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia del elemento i con el sistema del elemento $i - 1$. Las transformaciones básicas que relacionan el sistema de referencia del elemento i con el sistema del elemento $i - 1$ son:

- Rotación alrededor del eje Z_{i-1} un ángulo θ_i .
- Traslación a lo largo de Z_{i-1} una distancia d_i .
- Traslación a lo largo de X_i una distancia a_i .
- Rotación alrededor del eje X_i un ángulo α_i .

El algoritmo D-H se resume mediante la siguiente secuencia de pasos:

- Numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y acabando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.
- Numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y acabando en n.
- Localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.
- Para i de 0 a $n - 1$ situar el eje Z_i sobre el eje de la articulación $i+1$.
- Situar el origen del sistema de la base $\{S_0\}$ en cualquier punto del eje Z_0 . Los ejes X_0 , Y_0 se situarán de modo que formen un sistema dextrógiro con Z_0 .
- Para i de 1 a $n-1$, situar X_i en la línea normal común a Z_{i-1} y Z_i .
- Para i de 1 a $n-1$, situar Y_i de modo que forme un sistema dextrógiro con X_i y Z_i .
- Situar el sistema $\{S_n\}$ en el extremo del robot de modo que Z_n coincida con la dirección de Z_{n-1} y X_n sea normal a Z_{n-1} y Z_n .
- Obtener θ_i como el ángulo que hay que girar en torno a Z_{i-1} para que X_{i-1} y X_i queden paralelos.
- Obtener d_i como la distancia, medida a lo largo de Z_{i-1} , que habría que desplazar $\{S_{i-1}\}$ para que X_i y X_{i-1} quedasen alineados.
- Obtener a_i como la distancia medida a lo largo de X_i , que ahora coincidiría con X_{i-1} , que habría que desplazar el nuevo $\{S_{i-1}\}$ para que su origen coincidiese con $\{S_i\}$.
- Obtener α_i como el ángulo que habría que girar en torno a X_i , que ahora coincidiría con X_{i-1} , para que el nuevo $\{S_{i-1}\}$ coincidiese totalmente con $\{S_i\}$.
- Obtener las matrices de transformación

$${}^{i-1}A_i = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & a_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -s\alpha_i c\theta_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

- Obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot:

$$T = {}^0A_1 {}^1A_2 \dots {}^{n-1}A_n \quad (3)$$

- La matriz T define la orientación (sub-matriz de rotación) y posición (vector de traslación) del extremo referido a la base en función de las n coordenadas articulares.

1.4 Restricciones cinemáticas

Para robots que contienen una o más estructuras con cadenas cinemáticas cerradas, el cálculo del modelo geométrico requiere de la solución de ecuaciones geométricas de restricción por cada lazo cerrado. El objetivo es calcular las variables de las juntas pasivas en términos de las juntas variables activas.

1.4.1 Ecuaciones de cierre

En un lazo cinemático cerrado, existe una condición de cierre que se origina dado que los eslabones de una cadena cerrada no pueden abrirse cuando se mueven. Al expresar matemáticamente esa condición, se originan ecuaciones de restricción. Durante este capítulo se utilizará a manera de ejemplo ilustrativo simple, el mecanismo planar de cuatro barras de la Figura 2, el cual posee un GDL. El hecho de que estos cuatro eslabones siempre formen una cadena cerrada puede ser expresado matemáticamente por medio de las siguientes ecuaciones:

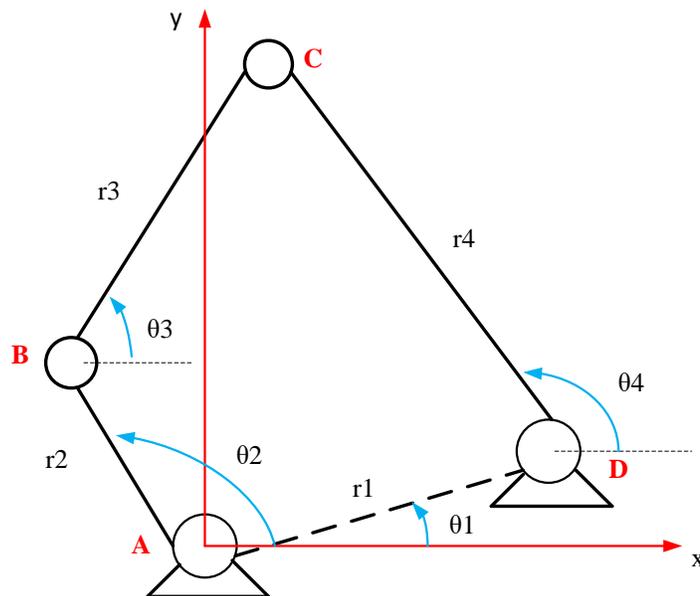


Figura 2. Mecanismo de cuatro barras

$$\vec{AB} + \vec{BC} + \vec{CD} + \vec{DA} = 0 \quad (4)$$

$$\begin{aligned}
r_1 \cos \theta_1 + r_2 \cos \theta_2 + r_3 \cos \theta_3 + r_4 \cos \theta_4 &= 0 \\
r_1 \sin \theta_1 + r_2 \sin \theta_2 + r_3 \sin \theta_3 + r_4 \sin \theta_4 &= 0 \\
\theta_1 + \theta_2 + \theta_3 + \theta_4 - 2\pi &= 0
\end{aligned} \tag{5}$$

Estas ecuaciones escalares se obtienen viendo el mecanismo de cuatro barras como una cadena cinemática serial con cuatro juntas de revolución, estas son denominadas ecuaciones de lazo cerrado. Tales restricciones son conocidas como restricciones holonómicas, las cuales pueden escribirse como una ecuación algebraica que involucra únicamente variables de posición pero no de velocidad.

1.4.1 Solución de las ecuaciones de restricción

Para dar solución a estas ecuaciones utilizamos el método descrito en [1]; se debe tener en cuenta que el ángulo de entrada θ_2 es conocido ya que está asociado a la articulación actuada y es manipulado por medio de un actuador, θ_1 se conoce de la geometría del mecanismo ya que es un ángulo constante perteneciente al eslabón fijo (tierra). El sistema de dos ecuaciones no lineales con dos incógnitas de la ecuación (5) se conoce como la ecuación de Freudenstein [4] y es usada en la síntesis de mecanismos. Su solución establece la relación cinemática entre la posición del eslabón de salida, con la posición del eslabón de entrada θ_2 .

El siguiente método permite resolver la ecuación de Freudenstein.

Considerando la siguiente expresión resultante de ordenar la ecuación (5):

$$\begin{aligned}
r_3 \cos \theta_3 &= r_1 \cos \theta_1 + r_4 \cos \theta_4 - r_2 \cos \theta_2 \\
r_3 \sin \theta_3 &= r_1 \sin \theta_1 + r_4 \sin \theta_4 - r_2 \sin \theta_2
\end{aligned} \tag{6}$$

Se elimina θ_3 elevando ambos lados de la ecuación (6) al cuadrado y sumando, llegando a:

$$\begin{aligned}
[r_3 \cos \theta_3]^2 + [r_3 \sin \theta_3]^2 &= [r_1 \cos \theta_1 + r_4 \cos \theta_4 - r_2 \cos \theta_2]^2 \\
&+ [r_1 \sin \theta_1 + r_4 \sin \theta_4 - r_2 \sin \theta_2]^2
\end{aligned}$$

Después de operar y organizar se obtiene:

$$\begin{aligned}
(2r_1r_4 \cos \theta_1 - 2r_2r_4 \cos \theta_2) \cos \theta_4 + (2r_1r_4 \sin \theta_1 \\
- 2r_2r_4 \sin \theta_2) \sin \theta_4 \\
- 2r_1r_2(\cos \theta_1 \cos \theta_2 + \sin \theta_1 \sin \theta_2) + r_1^2 + r_2^2 + r_4^2 - r_3^2 = 0
\end{aligned} \tag{7}$$

En la ecuación (7) sobrevive una incógnita, a saber, θ_4 . Dicha ecuación es de la forma de la ecuación (8), donde α es la incógnita a resolver.

$$A \cos \alpha + B \sin \alpha + C = 0 \quad (8)$$

La solución de α en la ecuación (8) se obtiene definiendo γ tal que,

$$\cos \gamma = \frac{A}{\sqrt{A^2+B^2}} \quad ; \quad \sin \gamma = \frac{B}{\sqrt{A^2+B^2}}$$

Con lo que,

$$\gamma = \text{atan2} \left(\frac{B}{\sqrt{A^2+B^2}}, \frac{A}{\sqrt{A^2+B^2}} \right) \quad (9)$$

Dividiendo los términos de la ecuación (8) por $\sqrt{A^2+B^2}$, el resultado puede ser escrito como:

$$\left(\cos \gamma \cos \alpha + \sin \gamma \sin \alpha + \frac{C}{\sqrt{A^2+B^2}} = 0 \right)$$

O:

$$\cos(\alpha - \gamma) + \frac{C}{\sqrt{A^2+B^2}} = 0$$

Con lo que se obtienen dos posibles soluciones para α en el intervalo $(-\pi, \pi]$ en términos del ángulo conocido γ :

$$\alpha = \gamma \pm \cos^{-1} \left(\frac{-C}{\sqrt{A^2+B^2}} \right) \quad (10)$$

Retornando a la solución de la ecuación (7), si se define:

$$A = (2r_1r_4\cos\theta_1 - 2r_2r_4\cos\theta_2)$$

$$B = (2r_1r_4\sin\theta_1 - 2r_2r_4\sin\theta_2)$$

$$C = (r_1^2 + r_2^2 + r_4^2 - r_3^2) - 2r_1r_2(\cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2)$$

Y γ como:

$$\gamma = \text{atan2} \left(\frac{2r_1r_4\sin\theta_1 - 2r_2r_4\sin\theta_2}{\sqrt{(2r_1r_4\cos\theta_1 - 2r_2r_4\cos\theta_2)^2 + (2r_1r_4\sin\theta_1 - 2r_2r_4\sin\theta_2)^2}}, \frac{2r_1r_4\cos\theta_1 - 2r_2r_4\cos\theta_2}{\sqrt{(2r_1r_4\cos\theta_1 - 2r_2r_4\cos\theta_2)^2 + (2r_1r_4\sin\theta_1 - 2r_2r_4\sin\theta_2)^2}} \right)$$

Entonces θ_4 es:

$$\theta_4 = \gamma \pm \cos^{-1} \left(\frac{-[(r_1^2 + r_2^2 + r_4^2 - r_3^2) - 2r_1r_2(\cos\theta_1\cos\theta_2 + \sin\theta_1\sin\theta_2)]}{\sqrt{(2r_1r_4\cos\theta_1 - 2r_2r_4\cos\theta_2)^2 + (2r_1r_4\sin\theta_1 - 2r_2r_4\sin\theta_2)^2}} \right) \quad (11)$$

La solución para θ_3 se obtiene dividiendo la segunda ecuación en (6) entre la primera ecuación en (6), obteniendo:

$$\theta_3 = \text{atan2} \left(\frac{r_1\sin\theta_1 + r_4\sin\theta_4 - r_2\sin\theta_2}{r_3}, \frac{r_1\cos\theta_1 + r_4\cos\theta_4 - r_2\cos\theta_2}{r_3} \right) \quad (12)$$

Nótese que tanto para θ_3 como para θ_4 existen dos posibles soluciones correspondientes a dos posibles configuraciones que puede adquirir el mecanismo (codo arriba o codo abajo), y dependiendo de la conveniencia geométrica para el mecanismo se elige ya sea la solución negativa o la positiva o posibles combinaciones de ambas, cuando se tratan de dos o más cadenas cerradas.

1.5 Modelo cinemático directo

El modelo cinemático directo describe las velocidades del efector final del robot en el espacio cartesiano ν , como función de las velocidades de las articulaciones, $\dot{\theta}$, expresadas en el espacio articular, siendo x el vector de posición del efector final en el espacio cartesiano y θ el vector de posición de las articulaciones en el espacio articular.

1.5.1 Cinemática de sistemas con cadenas cinemáticas cerradas

En esta sección, se considera la cinemática de sistemas robóticos con cadenas cerradas sujetas a restricciones cinemáticas en velocidad. Los conceptos

utilizados han sido extractados de [5]. Para tener en cuenta las restricciones cinemáticas derivadas de las ecuaciones de cierre, se extiende el método de cadenas seriales a cadenas cerradas abriendo el lazo de la cadena cerrada en un punto, propagando la cinemática del mecanismo a lo largo de sus ramas serie resultantes, y agregando las restricciones cinemáticas.

Las restricciones cinemáticas pueden ser representadas como restricciones algebraicas en el espacio de configuraciónes así:

$$c(\theta) = 0 \quad (13)$$

Siendo $c(\theta): \mathbb{R}^n \mapsto \mathbb{R}^r$ con n el número de articulaciones y r el número de restricciones, como se vio en el ejemplo de la ecuación (4). La ecuación (13) es un tipo de restricción holonómica.

Se define el número de grados de libertad efectivos del mecanismo n_e como:

$$n_e = n - r \quad (14)$$

Se asumirá que el número de juntas activas n_a debe ser igual a los grados de libertad efectivos, o $n_a = n - r$. Dado que el número de articulaciones pasivas $n_p = n - n_a$, se concluye que esta asunción implica que $n_p = r$.

La restricción puede ser escrita en términos de los vectores de velocidad de las juntas como:

$$J_C(\theta)\dot{\theta} = 0 \quad (15)$$

Donde $J_C = \frac{\partial c(\theta)}{\partial \theta} \in \mathbb{R}^{r \times n}$ es llamada la jacobiana de la restricción. Por otro lado la velocidad del efector final v compuesta por las velocidades lineales \dot{x} y angulares w , puede ser relacionada con la velocidad $\dot{\theta}$ por:

$$v = \begin{bmatrix} \dot{x} \\ w \end{bmatrix} = J_m(\theta)\dot{\theta} \quad (16)$$

Donde $J_m \in \mathbb{R}^{m \times n}$ es la jacobiana geométrica del manipulador.

El vector de variables articulares $\theta \in \mathbb{R}^n$ puede ser descompuesto como $\theta^T = [\theta_a^T \ \theta_p^T]$, donde $\theta_a \in \mathbb{R}^{n-r}$ es el vector de posiciones angulares de las juntas activas y $\theta_p \in \mathbb{R}^r$ el vector de posiciones angulares de las juntas pasivas.

Haciendo una partición de las jacobianas J_C y J_m de acuerdo a la dimensión de las variables de las juntas pasivas y activas se obtiene:

$$J_C = \begin{bmatrix} J_{Ca} & J_{Cp} \end{bmatrix} \quad (17)$$

$$J_m = \begin{bmatrix} J_{ma} & J_{mp} \end{bmatrix} \quad (18)$$

Entonces las ecuaciones (15) y (16) pueden reescribirse como:

$$0 = J_{Ca}\dot{\theta}_a + J_{Cp}\dot{\theta}_p \quad (19)$$

$$v = J_{ma}\dot{\theta}_a + J_{mp}\dot{\theta}_p \quad (20)$$

Donde $J_{Ca} \in \mathbb{R}^{r \times n_a}$, $J_{Cp} \in \mathbb{R}^{r \times n_p}$, $J_{ma} \in \mathbb{R}^{m \times n_a}$ y $J_{mp} \in \mathbb{R}^{m \times n_p}$, con m la dimensión del vector de velocidad del efector final. De la ecuación (19), $\dot{\theta}_p$ puede ser calculado en términos de las juntas activas como:

$$\dot{\theta}_p = -J_{Cp}^{-1}J_{Ca}\dot{\theta}_a \quad (21)$$

Donde J_{Cp} debe ser invertible (lo cual ocurriría solo si la asunción $n_p = r$ se cumple) y sustituyendo la ecuación (21) en la ecuación (20) se llega a la ecuación de la cinemática diferencial:

$$v = (J_{ma} - J_{mp}J_{Cp}^{-1}J_{Ca})\dot{\theta}_a = \bar{J}\dot{\theta}_a \quad (22)$$

Donde $\bar{J} \in \mathbb{R}^{m \times n_a}$ es la matriz jacobiana buscada que permite describir las velocidades del efector final como función de las velocidades articulares de las juntas activas.

Analizando la ecuación (22) se concluye que si el rango de la matriz J_{Cp} no es completo, ocurriría una singularidad en \bar{J} , en este caso habría un movimiento interno de las juntas, incluso cuando se bloquean las juntas activas.

1.5.2 Cinemática diferencial

La cinemática diferencial considerada para estos manipuladores, se calcula considerando las varias cadenas cinemáticas abiertas que componen la estructura

del mecanismo. La velocidad del efector final v puede ser obtenida a través de la derivada en el tiempo de las ecuaciones de la estructura del mecanismo, resultando en una matriz jacobiana para cada cadena cinemática serial. De esta manera, para nuestro sistema de cuatro barras tenemos:

$$v = SJ_1 \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = SJ_2 \begin{bmatrix} \dot{\theta}_2 \\ \dot{\theta}_4 \end{bmatrix} \quad (23)$$

$$S = \begin{bmatrix} & v_x & v_y & v_z & w_x & w_y & w_z \\ x & 1 & 0 & 0 & 0 & 0 & 0 \\ y & 0 & 1 & 0 & 0 & 0 & 0 \\ z & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (24)$$

Donde S es una matriz de selección que permite seleccionar las velocidades cartesianas o angulares de interés.

La jacobiana del mecanismo puede ser reescrita de una forma más usual tomando la jacobiana de cada cadena abierta serial así:

$$\begin{bmatrix} SJ_1 & 0 \\ 0 & SJ_2 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} = \begin{bmatrix} I \\ I \end{bmatrix} v \quad (25)$$

O de una forma equivalente:

$$J\dot{\theta} = Av \quad (26)$$

Donde A es: $\begin{bmatrix} I \\ I \end{bmatrix}$

Usando esta notación es posible obtener la jacobiana del manipulador por medio de:

$$J_m = A^\dagger J \quad (27)$$

Donde A^\dagger es la pseudo-inversa de A tal que $A^\dagger A = I$. Una posible opción es $A^\dagger = \begin{bmatrix} I & 0 \end{bmatrix}$.

1.6 Mecanismos sub-actuados

Un mecanismo sub-actuado es aquel mecanismo que posee menos actuadores que grados de libertad disponibles, de manera que al reducir el número de

actuadores se obtienen ventajas tales como: se reduce el consumo energético, los esfuerzos de control, su costo y se logran mecanismos de peso ligero. Los movimientos para este tipo de mecanismos pueden ser generados por la acción de actuadores, conocida como actuación directa, o bien por otros elementos a los que se encuentren interconectados, llamada actuación indirecta. En contraposición, estos mecanismos sub-actuados conducen a modelos cinemáticos y dinámicos complejos matemáticamente, los cuales hacen un poco difícil su análisis e implementación en simulación.

1.6.1 Diseño de un dedo antropomórfico sub-actuado

En esta subsección trataremos el diseño CAD de un mecanismo antropomórfico para dedo índice sub-actuado, el cual consta de tres características principales: movimiento antropomórfico, dimensiones geométricas apropiadas y un diseño de bajo costo.

Este diseño se basa en un mecanismo doble de 4 barras cruzadas, el cual proporciona un movimiento adecuado para nuestro análisis de posición de la yema de dedo índice. El diseño CAD se realizó en el software SolidWorks 2010 como se observa en la Figura 3 realizando un primer análisis de movimiento con el ensamblaje terminado.

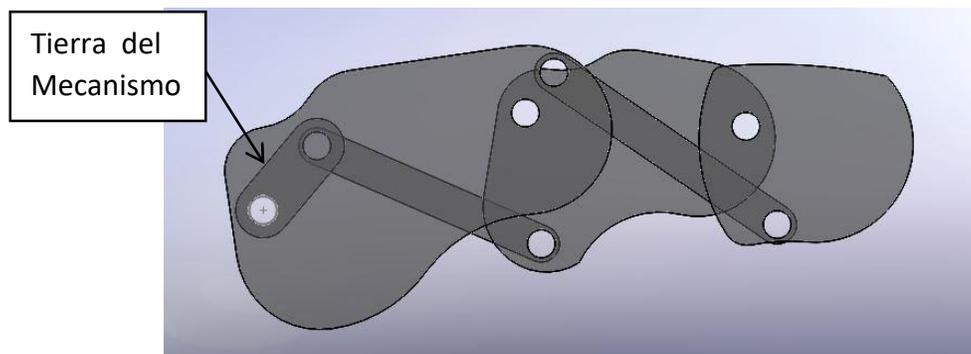


Figura 3. CAD dedo índice sub-actuado

1.6.2 Análisis de posición para el dedo

Aquí se presenta el análisis cinemático de posición para los dos lazos que conforman un mecanismo Stephenson I de seis barras. Este mecanismo de doble lazo, posee cuatro ramas para su análisis, dos para la parte de cuatro barras estándar y dos para las otras ramas resultantes del lazo como se ilustra en la Figura 4.

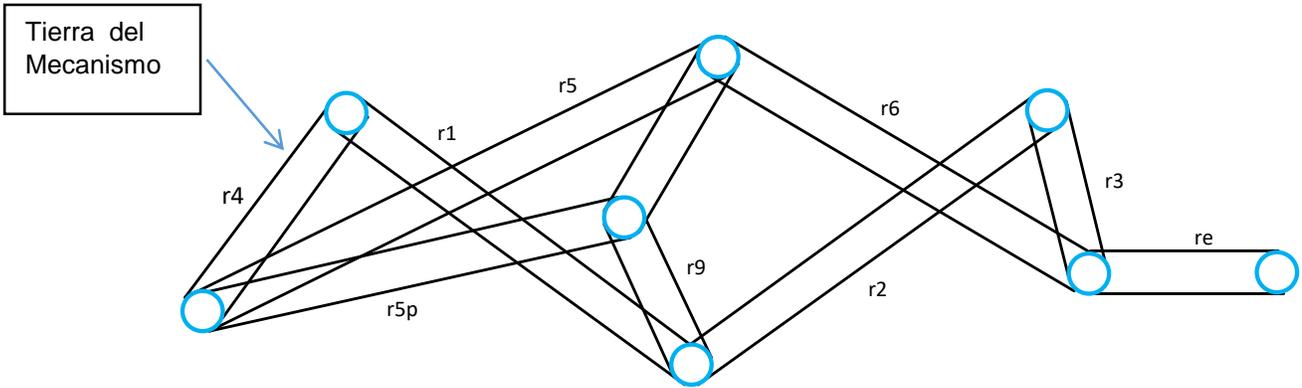


Figura 4. Mecanismo doble de 4 barras cruzadas

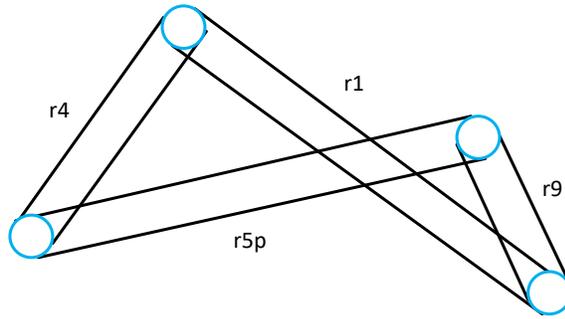


Figura 5. Mecanismo 4 barras cadena interna

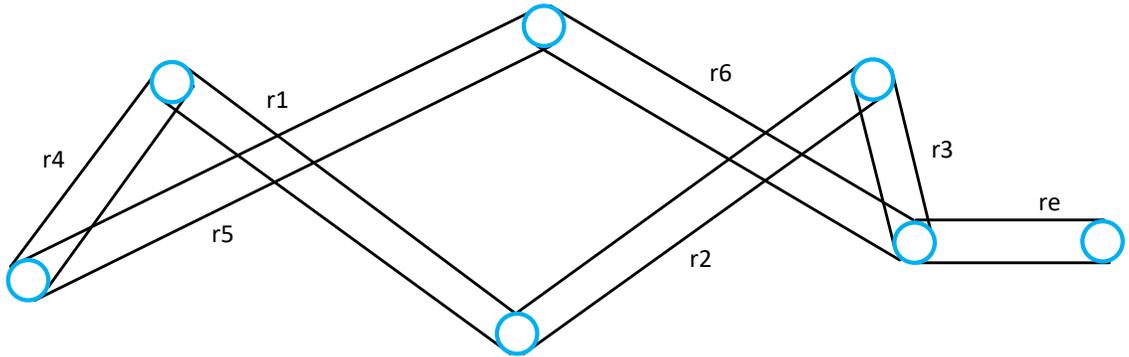


Figura 6. Cadena externa

Este mecanismo fue el utilizado para la implementación del dedo índice, cuyas medidas de sus falanges se presentan en la siguiente tabla y se visualizan en las Figuras 7,8 y 9. La tierra del mecanismo como se ilustra en la Figura 3 y 4 es r4.

Falange proximal (mm)	Falange media (mm)	Falange distal (mm)
34.54	25.27	19.87

Tabla 1. Dimensiones de las Falanges del dedo sub-actuado

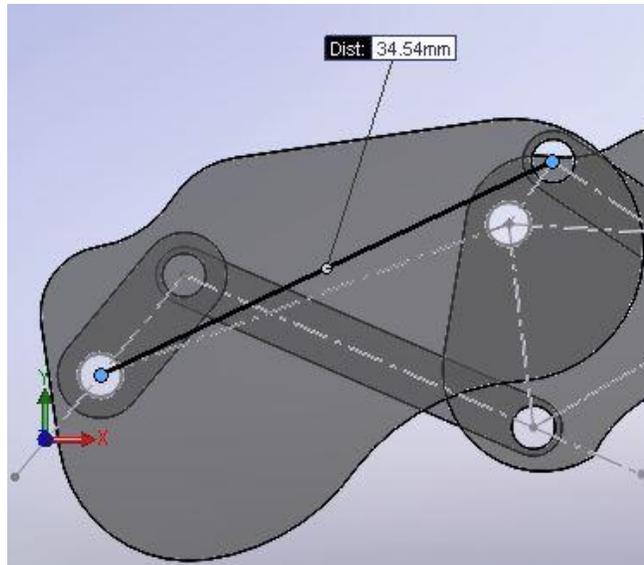


Figura 7. Falange Proximal

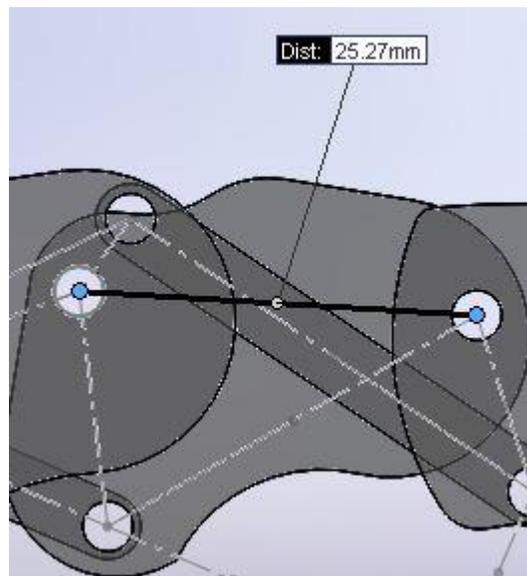


Figura 8. Falange Media

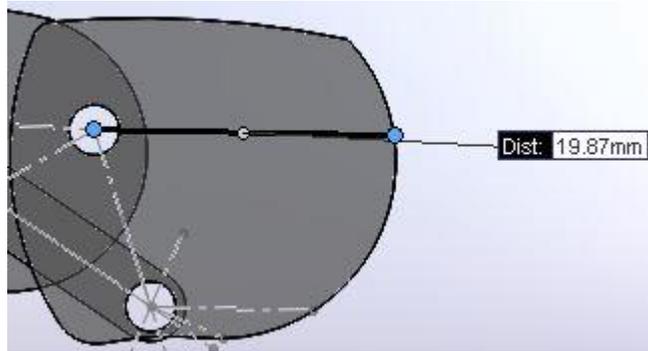


Figura 9. Falange Distal

Para el análisis cinemático de posición se empleó el formalismo D-H descrito en la sección 1.3.1 para la construcción del modelo geométrico del dedo sub-actuado, el cual relaciona los parámetros geométricos entre eslabones por medio de matrices de transformación.

Teniendo en cuenta que no se trata con cadenas cinemáticas seriales (abiertas) si no, con cadenas cinemáticas cerradas con restricciones, se abren los lazos cerrados en una de sus juntas pasivas y se propaga la cinemática por cada una de las ramas seriales que resultan al abrir estos lazos como se indicó en la sección 1.5.1. Como herramienta de análisis se utilizó el toolbox de MATLAB para robótica desarrollado por Peter Corke, el cual proporciona funciones básicas para el estudio y simulación de sistemas robóticos seriales. Este toolbox permite desarrollar el modelo geométrico directo para encontrar las ecuaciones de restricción a partir de los parámetros geométricos del dedo sub-actuado, donde la única articulación actuada es θ_1 (th1) las demás son pasivas θ_2 (th2), θ_3 (th3), θ_4 (th4), θ_5 (th5), θ_6 (th6), las distancias de los eslabones serán constantes ($r_1, r_2, r_3, r_4, r_5, r_{5p}, r_6, r_7, r_9, r_e$), al igual que los parámetros θ_4 (th4), α (alf) y β (bet). En la Figura 10 se ilustra lo descrito anteriormente.

Siguiendo la metodología D-H, se ubicó el sistema de coordenadas correspondiente a un sistema dextrógiro siguiendo la regla de la mano derecha, los ejes Z de las articulaciones rotoides saldrán del papel. Los ejes x se ubicaron a lo largo de los eslabones, siendo perpendicularmente comunes a los ejes Z_{j-1} y Z_j de cada articulación.

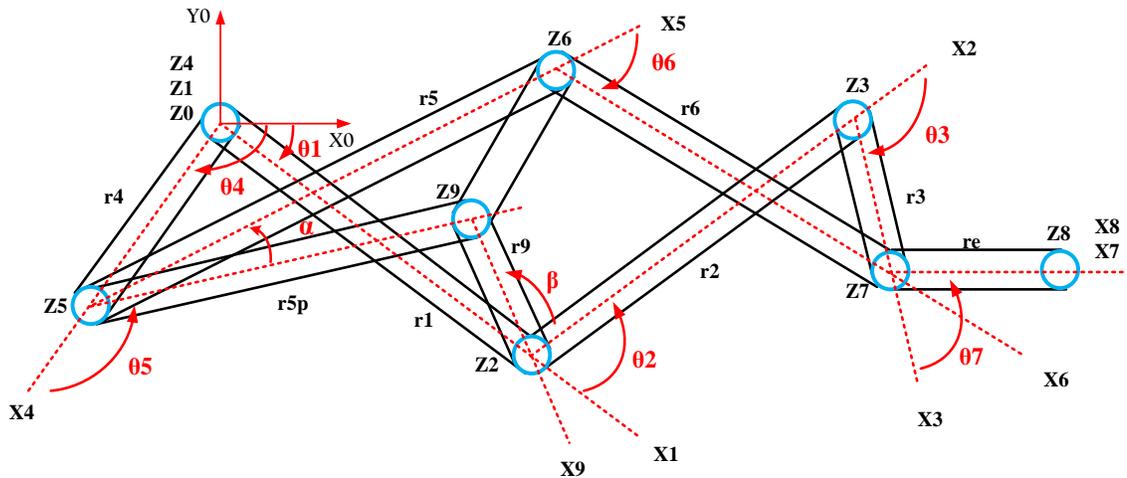


Figura 10. Modelo del dedo con los marcos coordenados D-H.

1.6.2.1 Parámetros D-H para la cadena cinemática abierta 1

En la Figura 11 se presenta la definición de los parámetros D-H para la primera cadena cinemática abierta resultante de abrir la cadena cinemática cerrada de la Figura 5 en la unión de los enlaces r_1 - r_9 . Considerando que todas las juntas son de revolución, el parámetro sigma es 0 el cual al definir la junta como revolvente (R) automáticamente este valor se asigna como 0. Si este no fuera el caso y fuese una articulación prismática, su valor sería 1. Además, dado que los ejes Z de las articulaciones se ubican en paralelo entre sí sobre el mismo plano, los parámetros d y alpha son 0. El offset es una constante de desplazamiento entre el vector de ángulo de la junta del usuario y la solución cinemática verdadera, es útil si se desea que el robot adopte una postura "sensible" para la configuración de ángulo de junta cero. En este caso es 0.

```

RL1int =
RL1int (2 axis, RR, stdDH, slowRNE)
+-----+
| j |   theta |     d |     a |   alpha |   offset |
+-----+
| 1 |     q1 |     0 |    r1 |     0 |     0 |
| 2 |     q2 |     0 |    r9 |     0 |     0 |
+-----+

grav =   0   base = 1 0 0 0   tool = 1 0 0 0
        0       0 1 0 0       0 1 0 0
        9.81    0 0 1 0       0 0 1 0
                0 0 0 1       0 0 0 1

```

Figura 11. Parámetros D-H cadena cinemática abierta 1.

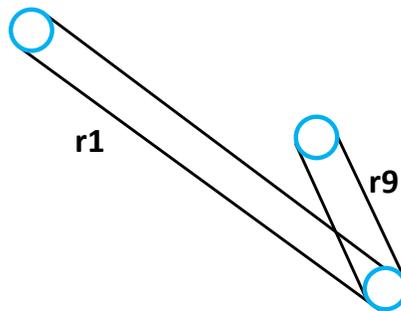


Figura 12. Cadena cinemática abierta 1.

1.6.2.2 Parámetros D-H para la cadena cinemática abierta 2

En la Figura 13 se presenta la definición de los parámetros D-H para la segunda cadena cinemática abierta resultante de abrir la cadena cinemática cerrada de la Figura 5 en la unión de los enlaces r1-r9. Dado que r4 es un enlace de tierra, el robot resultante es de un solo enlace, a saber, r5p. En este caso se define la matriz de transformación de la base del sistema, la cual tiene una longitud r4 rotada un ángulo fijo th4 medido desde el marco de referencia. La Figura 14 muestra esta cadena independiente.

```

RL2int =

RL2int (1 axis, R, stdDH, slowRNE)

+-----+-----+-----+-----+-----+
| j |   theta |     d |     a |   alpha |  offset |
+-----+-----+-----+-----+-----+
| 1 |     q |     0 |  r5p |     0 |     0 |
+-----+-----+-----+-----+-----+

grav =   0   base = cos(th4) -sin(th4)  0   r4*cos(th4)   tool = 1  0  0  0
          0         sin(th4)  cos(th4)  0   r4*sin(th4)       0  1  0  0
          9.81      0         0         1  0                0  0  1  0
                   0         0         0  1                0  0  0  1

```

Figura 13. Parámetros D-H cadena cinemática abierta 2

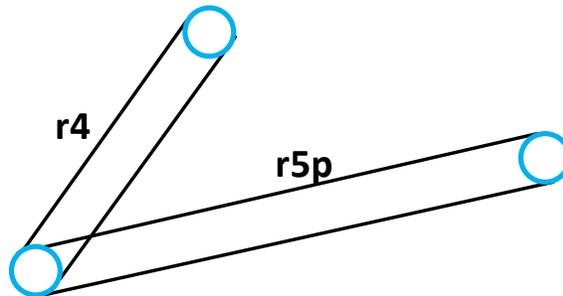


Figura 14. Cadena cinemática abierta 2.

1.6.2.3 Parámetros D-H para la cadena cinemática abierta 3

En la Figura 15 se presenta la definición de los parámetros D-H para la tercera cadena cinemática abierta, resultante de abrir la cadena cinemática cerrada de la Figura 6 en la unión de los enlaces r_3 - r_6 . En este caso se define la matriz de transformación de la herramienta del robot (efector final), la cual será la yema del dedo sub-actuado, definida como una transformación de longitud r_e , rotada un ángulo fijo th_7 , como se ilustra en la Figura 16.

```

RLiext =

RLiext (3 axis, RRR, stdDH, slowRNE)

+-----+-----+-----+-----+-----+
| j |   theta |     d |     a |   alpha |  offset |
+-----+-----+-----+-----+-----+
| 1 |    q1 |     0 |    r1 |     0 |     0 |
| 2 |    q2 |     0 |    r2 |     0 |     0 |
| 3 |    q3 |     0 |    r3 |     0 |     0 |
+-----+-----+-----+-----+-----+

grav =   0 base = 1 0 0 0 tool =  cos(th7) -sin(th7) 0 re*cos(th7)
          0           0 1 0 0       sin(th7)  cos(th7) 0 re*sin(th7)
          9.81        0 0 1 0         0         0     1 0
          0 0 0 1         0         0     0 1

```

Figura 15. Parámetros D-H cadena cinemática abierta 3

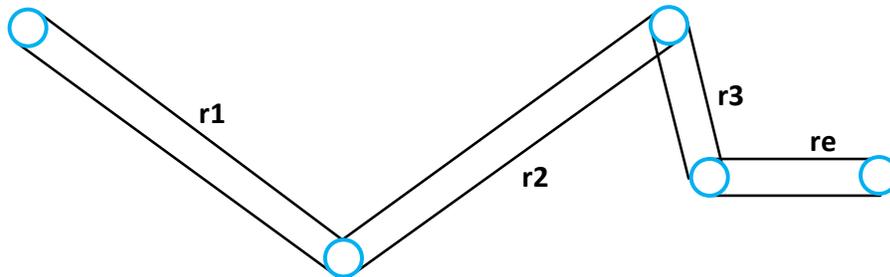


Figura 16. Cadena cinemática abierta 3.

1.6.2.4 Parámetros D-H para la cadena cinemática 4

En la Figura 17 se presenta la definición de los parámetros D-H para la cuarta cadena cinemática abierta, resultante de abrir la cadena cinemática cerrada de la Figura 6 en la unión de los enlaces r3-r6.

RL2ext =

RL2ext (2 axis, RR, stdDH, slowRNE)

j	theta	d	a	alpha	offset
1	q1	0	r5	0	0
2	q2	0	r6	0	0

```

grav = 0 base = cos(th4) -sin(th4) 0 r4*cos(th4) tool = cos(th7) -sin(th7) 0 re*cos(th7)
          0          sin(th4) cos(th4) 0 r4*sin(th4)      sin(th7) cos(th7) 0 re*sin(th7)
          9.81       0          0          1 0              0          0          1 0
                   0          0          0 1              0          0          0 1
  
```

Figura 17. Parámetros D-H cadena cinemática abierta 4.

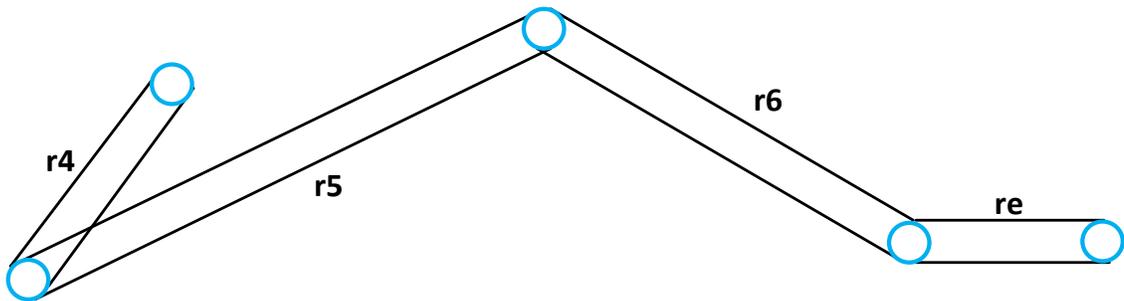


Figura 18. Cadena cinemática abierta 4.

1.6.2.5 Cinemática directa para la cadena abierta 1

La matriz de transformación para posición y orientación del punto de apertura de la cadena abierta 1, respecto a la base, se muestra en la Figura. 19. Contiene una sub-matriz de rotación y un vector de posición, este último es de interés ya que permite describir la ecuación de restricción de lazo cerrado para esta cadena. Así, para la cadena 1 la cuarta columna de $T1_{int}$ es el vector de posición correspondiente al punto de apertura respecto a la base.

```
T1int =
[ cos(bet + th1 + th2), -sin(bet + th1 + th2), 0, r1*cos(th1) + r9*cos(bet + th1 + th2)]
[ sin(bet + th1 + th2),  cos(bet + th1 + th2), 0, r1*sin(th1) + r9*sin(bet + th1 + th2)]
[      0,      0, 1, 0]
[      0,      0, 0, 1]
```

Figura 19. Cinemática directa cadena abierta 1.

1.6.2.6 Cinemática directa para la cadena abierta 2

La matriz de transformación para posición y orientación del punto de apertura de la cadena abierta 2, respecto a la base, se muestra en la Figura 20. El vector de posición del punto de apertura es de interés ya que permite establecer la ecuación de restricción de lazo cerrado de esta cadena cinemática. La cuarta columna de T2int es el vector de posición correspondiente.

```
T2int =
[ cos(alf - th4 - th5), sin(alf - th4 - th5), 0, r5p*cos(alf - th4 - th5) + r4*cos(th4)]
[ -sin(alf - th4 - th5), cos(alf - th4 - th5), 0, r4*sin(th4) - r5p*sin(alf - th4 - th5)]
[      0,      0, 1, 0]
[      0,      0, 0, 1]
```

Figura 20. Cinemática directa cadena abierta 2.

1.6.2.7 Cinemática directa para la cadena abierta 3

La matriz de transformación para posición y orientación del punto de apertura de la cadena abierta 3, respecto a la base, se muestra en la Figura 21. La cuarta columna de T1ext es el vector de posición, a partir del cual se puede describir una restricción para la cadena cinemática cerrada.

```
T1ext =
[ cos(th1 + th2 + th3 + th7), -sin(th1 + th2 + th3 + th7), 0, re*cos(th1 + th2 + th3 + th7) + r2*cos(th1 + th2) + r1*cos(th1) + r3*cos(th1 + th2 + th3)]
[ sin(th1 + th2 + th3 + th7),  cos(th1 + th2 + th3 + th7), 0, r3*sin(th1 + th2 + th3) + re*sin(th1 + th2 + th3 + th7) + r2*sin(th1 + th2) + r1*sin(th1)]
[      0,      0, 1, 0]
[      0,      0, 0, 1]
```

Figura 21. Cinemática directa cadena abierta 3.

1.6.2.8 Cinemática directa para la cadena 4

La matriz de transformación para posición y orientación del punto de apertura de la cadena abierta 4, respecto a la base, se muestra en la Figura 22. La cuarta columna de T2ext es el vector de posición, a partir del cual se puede describir una restricción para la cadena cinemática cerrada.

```
T2ext =
[ cos(th4 + th5 + th6 + th7), -sin(th4 + th5 + th6 + th7), 0, r2*cos(th4 + th5 + th6 + th7) + r5*cos(th4 + th5) + r4*cos(th4) + r6*cos(th4 + th5 + th6)]
[ sin(th4 + th5 + th6 + th7),  cos(th4 + th5 + th6 + th7), 0, r6*sin(th4 + th5 + th6) + r2*sin(th4 + th5 + th6 + th7) + r5*sin(th4 + th5) + r4*sin(th4)]
[      0,      0,      0, 1]
[      0,      0,      0, 0]
```

Figura 22. Cinemática directa cadena abierta 4.

1.6.3 Ecuaciones de restricción para el dedo sub-actuado

Las ecuaciones de restricción imponen la condición de lazo cerrado para las cadenas cinemáticas cerradas. A continuación se usan los conceptos descritos en la sección 1.6.2 para obtener el vector de restricciones holonómicas en el dedo robótico.

Se obtienen cuatro ecuaciones de restricción, resultantes de igualar las posiciones en x y en y de los puntos de apertura de las cadenas abiertas 1-2 y 3-4. Estas ecuaciones definen que esos puntos están restringidos a permanecer unidos.

Imponiendo la unión de los puntos de apertura de las cadenas 1 y 2 se obtiene:

$$0 = r9 \cos(th1 + th2p) + r1 \cos(th1) - r5p \cos(th4 + th5p) - r4 \cos(th4) \quad (28)$$

$$0 = r9 \sin(th1 + th2p) + r1 \sin(th1) - r5p \sin(th4 + th5p) - r4 \sin(th4)$$

Imponiendo la unión de los puntos de apertura de las cadenas 3 y 4 se obtiene:

$$\begin{aligned}
0 &= r_2 \cos(th_1 + th_2) + r_1 \cos(th_1) + r_3 \cos(th_1 + th_2 + th_3) \\
&\quad - r_5 \cos(th_4 + th_5) - r_4 \cos(th_4) \\
&\quad - r_6 \cos(th_4 + th_5 + th_6) \\
0 &= r_3 \sin(th_1 + th_2 + th_3) + r_2 \sin(th_1 + th_2) + r_1 \sin(th_1) \quad (29) \\
&\quad - r_6 \sin(th_4 + th_5 + th_6) - r_5 \sin(th_4 + th_5) \\
&\quad - r_4 \sin(th_4)
\end{aligned}$$

1.6.4 Solución de las ecuaciones de restricción

El método descrito en la sección 1.4.1 se aplica a continuación para solucionar las ecuaciones de restricción.

De la cadena interna se tiene como términos conocidos los ángulos θ_1, θ_4 y desconocidos θ_2, θ_5 además $\theta_{2p} = \theta_2 + \beta$ y $\theta_{5p} = \theta_5 - \alpha$.

Eliminando θ_{5p} tenemos:

$$\begin{aligned}
r_5 p \cos(\theta_4 + \theta_{5p}) \\
&= r_1 \cos(th_1) + r_9 \cos(th_1 + th_{2p}) - r_4 \cos(th_4) \quad (30) \\
r_5 p \sin(\theta_4 + \theta_{5p}) &= r_1 \sin(th_1) + r_9 \sin(th_1 + th_{2p}) - r_4 \sin(th_4)
\end{aligned}$$

Elevando al cuadrado ambos lados de la ecuación, sumando y simplificando, la expresión resultante es:

$$\begin{aligned}
r_5 p^2 &= r_1^2 + r_9^2 + r_4^2 + (2r_1 r_9 \cos \theta_1 - 2r_9 r_4 \cos \theta_4) \cos(\theta_1 + \theta_{2p}) \\
&\quad + (2r_1 r_9 \sin \theta_1 - 2r_9 r_4 \sin \theta_4) \sin(\theta_1 + \theta_{2p}) - 2r_1 r_4 (\cos \theta_1 \cos \theta_4 \\
&\quad \quad \quad + \sin \theta_1 \sin \theta_4) \quad (31)
\end{aligned}$$

Los valores de A, B y C son:

$$\begin{aligned}
A &= 2r_1 r_9 \cos \theta_1 - 2r_9 r_4 \cos \theta_4 \\
B &= 2r_1 r_9 \sin \theta_1 - 2r_9 r_4 \sin \theta_4 \\
C &= r_1^2 + r_9^2 + r_4^2 - r_5 p^2 - 2r_1 r_4 (\cos \theta_1 \cos \theta_4 + \sin \theta_1 \sin \theta_4)
\end{aligned}$$

Aplicando ecuación (9) y ecuación (10):

$$\theta_{2p} = \gamma \pm \cos^{-1} \left(\frac{-C}{\sqrt{A^2 + B^2}} \right) - \theta_1$$

$$\begin{aligned} & \theta_{2p} \\ & = \gamma \pm \cos^{-1} \left(\frac{-(r_1^2 + r_9^2 + r_4^2 - r_5 p^2 - 2r_1 r_4 (\cos \theta_1 \cos \theta_4 + \sin \theta_1 \sin \theta_4))}{\sqrt{(2r_1 r_9 \cos \theta_1 - 2r_9 r_4 \cos \theta_4)^2 + (2r_1 r_9 \sin \theta_1 - 2r_9 r_4 \sin \theta_4)^2}} \right) \\ & - \theta_1 \end{aligned} \quad (32)$$

$$\gamma = \operatorname{atan2} \left(\frac{\frac{2r_1 r_9 \sin \theta_1 - 2r_9 r_4 \sin \theta_4}{\sqrt{(2r_1 r_9 \cos \theta_1 - 2r_9 r_4 \cos \theta_4)^2 + (2r_1 r_9 \sin \theta_1 - 2r_9 r_4 \sin \theta_4)^2}}}{\frac{2r_1 r_9 \cos \theta_1 - 2r_9 r_4 \cos \theta_4}{\sqrt{(2r_1 r_9 \cos \theta_1 - 2r_9 r_4 \cos \theta_4)^2 + (2r_1 r_9 \sin \theta_1 - 2r_9 r_4 \sin \theta_4)^2}}} \right)$$

Substituyendo θ_{2p} en la ecuación (32):

$$\theta_{5p} = \operatorname{atan2} \left(\frac{\frac{r_1 \sin \theta_1 + r_9 \sin(\theta_1 + \theta_{2p}) - r_4 \sin \theta_4}{r_5 p}, \frac{r_1 \cos \theta_1 + r_9 \cos(\theta_1 + \theta_{2p}) - r_4 \cos \theta_4}{r_5 p}}{-\theta_4} \right) \quad (33)$$

Ahora de la cadena externa, ecuación (31), quedan como incógnitas las variables θ_3 y θ_6 , así que se tiene un sistema de dos ecuaciones con dos incógnitas y de manera análoga se resuelve este set de ecuaciones.

Eliminando θ_6

$$\begin{aligned} & r_6 \cos(\theta_4 + \theta_5 + \theta_6) \\ & = r_1 \cos \theta_1 + r_3 \cos(\theta_1 + \theta_2 + \theta_3) + r_2 \cos(\theta_1 + \theta_2) \\ & - r_4 \cos \theta_4 - r_5 \cos(\theta_4 + \theta_5) \\ & r_6 \sin(\theta_4 + \theta_5 + \theta_6) \\ & = r_1 \sin \theta_1 + r_3 \sin(\theta_1 + \theta_2 + \theta_3) + r_2 \sin(\theta_1 + \theta_2) \\ & - r_4 \sin \theta_4 - r_5 \sin(\theta_4 + \theta_5) \end{aligned} \quad (34)$$

Haciendo:

$$D1 = r2 \cos(\theta_1 + \theta_2) - r4 \cos \theta_4 - r5 \cos(\theta_4 + \theta_5)$$

$$D2 = r2 \sin(\theta_1 + \theta_2) - r4 \sin \theta_4 - r5 \sin(\theta_4 + \theta_5)$$

Elevando al cuadrado ambos lados de la ecuación, sumando y reemplazando:

$$\begin{aligned} r6^2 = & r1^2 + r3^2 + (2r1r3 \cos \theta_1 + 2r3D1) \cos(\theta_1 + \theta_2 + \theta_3) \\ & + 2r1 \cos \theta_1 D1 + D1^2 \\ & + (2r1r3 \sin \theta_1 + 2r3D2) \sin(\theta_1 + \theta_2 + \theta_3) + 2r1 \sin \theta_1 D2 \\ & + D2^2 \end{aligned} \quad (35)$$

Los valores de A, B y C son:

$$A = 2r1r3 \cos \theta_1 + 2r3D1$$

$$B = 2r1r3 \sin \theta_1 + 2r3D2$$

$$C = r1^2 + r3^2 - r6^2 + D1^2 + 2r1 \cos \theta_1 D1 + D2^2 + 2r1 \sin \theta_1 D2$$

$$\theta_3 = \gamma_2 \pm \cos^{-1} \left(\frac{-(r1^2 + r3^2 - r6^2 + D1^2 + 2r1 \cos \theta_1 D1 + D2^2 + 2r1 \sin \theta_1 D2)}{\sqrt{(2r1r3 \cos \theta_1 + 2r3D1)^2 + (2r1r3 \sin \theta_1 + 2r3D2)^2}} \right) - \theta_1 - \theta_2 \quad (36)$$

$$\gamma_2 = \operatorname{atan2} \left(\frac{2r1r3 \sin \theta_1 + 2r3D2}{\sqrt{(2r1r3 \cos \theta_1 + 2r3D1)^2 + (2r1r3 \sin \theta_1 + 2r3D2)^2}}, \frac{2r1r3 \cos \theta_1 + 2r3D1}{\sqrt{(2r1r3 \cos \theta_1 + 2r3D1)^2 + (2r1r3 \sin \theta_1 + 2r3D2)^2}} \right)$$

Substituyendo θ_3 en la ecuación (34),

$$\theta_6 = \operatorname{atan2} \left(\frac{r1 \sin \theta_1 + r3 \sin(\theta_1 + \theta_2 + \theta_3) + D2}{r6}, \frac{r1 \cos \theta_1 + r3 \cos(\theta_1 + \theta_2 + \theta_3) + D1}{r6} \right) - \theta_4 - \theta_5 \quad (37)$$

1.7 Análisis cinemático dedo sub-actuado

Para el análisis de cinemática diferencial del dedo índice, se aplica lo expuesto en la subsección 1.5.2 a partir de las ecuaciones de restricción de lazo cerrado.

La jacobiana de la restricción se obtiene derivando parcialmente las ecuaciones (28) y (29) respecto a la variable actuada θ_1 y las variables pasivas $\theta_2, \theta_3, \theta_5, \theta_6$ esto sería $J_c = [J_{ca}, J_{cp}]$, sus dimensiones son: $J_c \in \mathbb{R}^{4 \times 5}, J_{ca} \in \mathbb{R}^{4 \times 1}, J_{cp} \in \mathbb{R}^{4 \times 4}$.

La jacobiana del manipulador se obtiene a partir de abrir la cadena cinemática cerrada externa, con lo cual existen dos caminos posibles correspondientes a las dos cadenas cinemáticas abiertas resultantes.

Sean J_1 y J_2 las matrices jacobianas del manipulador para las cadenas cinemáticas abiertas 3 y 4 respectivamente (obtenidas con el Toolbox RCV de MATLAB), y siendo la matriz de selección $S \in \mathbb{R}^{2 \times 6}$ dado que el vector de velocidad del efector final solo tiene componentes en x e y , el procedimiento descrito en la subsección 1.5.2 lleva a obtener la jacobiana del manipulador $J_m = [J_{ma}, J_{mp}]$.

La jacobiana buscada tal que la velocidad del efector final pueda describirse como función de la velocidad de la articulación activa, $v = \bar{J}\dot{\theta}_a$ será:

$$\bar{J} = (J_{ma} - J_{mp}J_{cp}^{-1}J_{ca}) \quad (38)$$

Cuya dimensión es $\bar{J} \in \mathbb{R}^{2 \times 1}$. Debido a la gran cantidad de términos que aparecen en esta jacobiana, su resultado no se muestra en esta monografía, pero puede ser consultada en el código de MATLAB generado que se anexa en CD, donde se obtiene un resultado explícito simbólico para esta matriz.

Con el fin de validar los resultados obtenidos en esta sección, se realizaron dos experimentos en MATLAB que se describen a continuación.

1. Para validar que los ángulos solución de la cinemática directa obtenidos en las ecuaciones (32), (33), (36) y (37) son correctos, se graficaron en MATLAB los enlaces del dedo a partir de líneas en un plano. Para un ángulo actuado θ_1 variando en el rango $-90^\circ < \theta_1 < 0^\circ$, se obtuvieron mil pasos de ángulo en ese rango para los cuales se calculó la solución de la cinemática y se rotaron los ángulos de las articulaciones pasivas en consecuencia. El resultado de este experimento se ilustra en la Figura 23, donde se muestran los enlaces al final del recorrido, y con un asterisco la

posición de la yema del dedo predicha por esa solución en cada paso de la trayectoria. El experimento reveló que para todo paso simulado, los enlaces de las dos cadenas cinemáticas cerradas permanecían sin abrirse, con lo cual se verifica que la solución (explícita en forma simbólica) obtenida, es correcta. También el experimento permitió seleccionar, de las dos posibles soluciones de cada ángulo pasivo, cuál era el signo correcto que debía ser seleccionado para que el dedo permaneciera en una configuración física correcta.

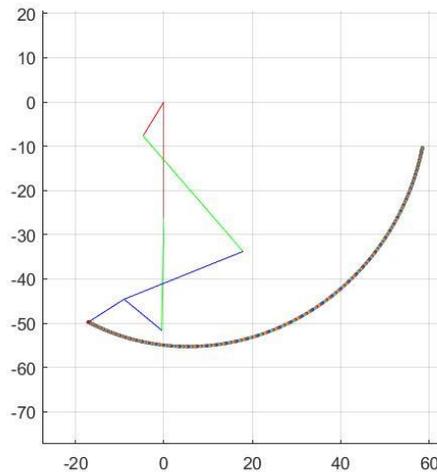


Figura 23. Experimento en MATLAB validación de ángulos

2. Generando para θ_1 una trayectoria grado 5, para ir desde la posición inicial 0° a la posición final -90° , se obtuvieron trayectorias de posición y velocidad angular de mil pasos discretos con período de muestreo de 1 milisegundo. Derivando el vector de posición de la yema respecto al tiempo (calculada con la solución de la cinemática directa), se obtienen las velocidades cartesianas de la yema de la Figura 24 representadas con color verde. De otro lado, dada la trayectoria de velocidad de la articulación activa generada, se calcula el mismo vector de velocidad cartesiana de la yema pero ahora predicho a través de la matriz Jacobiana de la ecuación (38) y que se representa con el trazo en color rojo de la Figura 24. Para mejor comparación, la Figura 24 también muestra la diferencia entre las velocidades calculadas por los dos métodos y que se describe como error de velocidad. La existencia de este error (aunque pequeño) se explica porque la derivada de la posición respecto al tiempo se hizo con el comando diff de MATLAB, el cual permite obtener solo un aproximado de la derivada no muy preciso. Sin embargo, la coincidencia en forma y en valores obtenida, permite asegurar que la jacobiana calculada es correcta.

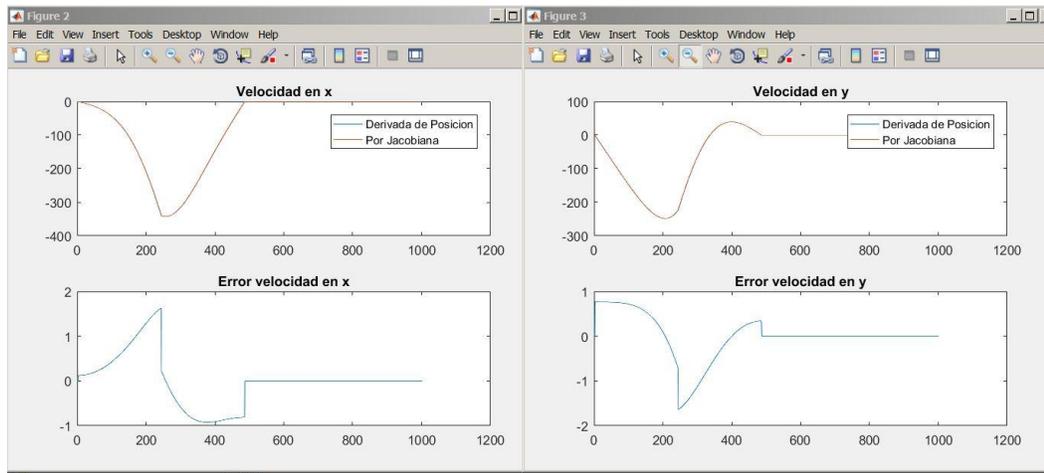


Figura 24. Grafica de la jacobiana del dedo en MATLAB

2 Representación del modelo dinámico usando Adams

En este capítulo se comprenderá la simulación mecánica de sistemas usando Adams, para el análisis dinámico de nuestro dedo índice sub-actuado. Adams (Automatic Dynamic Analysis of Mechanical Systems) tiene sus orígenes en la tesis de doctorado escrita en la universidad de Michigan en el año 1973 por el Dr. Nicolae Orlandea [6], la cual se basa en la dinámica de sistemas multicuerpo (acrónimo MBD). Se caracteriza sobre todo por su capacidad de ocuparse eficientemente del movimiento apreciable y en especial el movimiento rotacional que tiende a ser altamente no lineal. Estos efectos se agravan con el tiempo, a medida que varía según la no linealidad, por tanto un problema de MBD es matemáticamente no lineal. Adams [8-9] está específicamente diseñado para tratar eficientemente con estas no linealidades.

2.1 Generación del modelo en Adams

El software de Adams, trabaja generando un modelo dinámico de la planta, que se construye a partir de un sistema mecánico previamente diseñado y editado en Adams. Este modelo representara y simulara la dinámica presente en el sistema y proporcionara una herramienta de simulación grafica valiosa para el análisis del comportamiento de los sistemas de control y de la sensórica presente. Los sensores que incorpora Adams hacen posible que el sistema de control trabaje en lazo cerrado, lo cual genera un incremento en la precisión para seguir una señal de entrada, reduce la sensibilidad a variaciones en el modelo y el efecto de no linealidades.



Figura 24. Representación entrada-salida de Adams

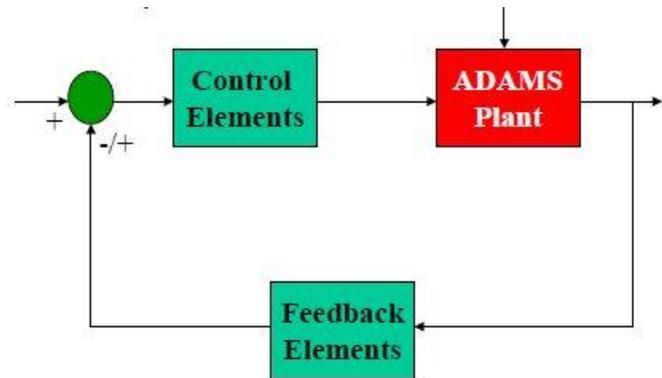


Figura 25. Representación en lazo cerrado de un sistema usando Adams

Para la generación del modelo dinámico en Adams se procede de la siguiente manera:

- Creación de un nuevo modelo en Adams, definiendo el directorio de trabajo, las unidades con las que se requieren trabajar y el sentido de la gravedad.
- Importación del modelo CAD ensamblado previamente en SolidWorks 2010 con extensión .lges
- Adición de propiedades físicas como lo son: masa de las piezas, momentos de inercia, fricciones en las articulaciones, fuerzas de contacto para simular colisiones con objetos, valor de la fuerza de gravedad y su sentido.
- Definición de las juntas de revolución, ya que al ser importado a Adams, Solidworks no mantiene las relaciones de posición del ensamble.
- Creación de marcadores para formar los sensores de ángulo que existen en el ambiente de Adams, siguiendo el algoritmo D-H y sensores para la medición de la posición del efector final (yema).
- Definición de los sensores generados con el paso anterior.
- Creación de un componente de fuerza aplicada (Torque), que se le asignara a la articulación actuada.
- Creación de variables de estado. Una variable de estado es una variable cuyo valor se calcula en cada paso de simulación, las cuales permiten realizar una manipulación externa por parte de Simulink. Adams Controls utiliza variables de estado para especificar las entradas y salidas del modelo Adams, porque estas variables de estado se calculan durante el tiempo de ejecución. Para nuestro caso estas se crearan para la medición de los ángulos de las articulaciones, la posición del efector final y la fuerza aplicada por parte del sistema de control.
- Exportación del modelo de la planta en Adams, quien tendrá inmersa la dinámica definida por los pasos anteriores, esto se realiza gracias al plugin de Adams Control, quien permite exportar un modelo no lineal o lineal de la planta a MATLAB, indicando las señales que serán sus entradas (actuadores) y las señales que serán salida del sistema (sensores).
- Por ultimo Adams genera una serie de archivos de MATLAB en el directorio donde se guardó el proyecto, una vez aplicados los cambios del paso anterior. En MATLAB debemos cargar toda la información de las variables y esto se hace escribiendo en la ventana de comandos el nombre con el cual se exporto la planta; por defecto, Adams crea nombres como Controls_Plant_#, donde el numeral hace referencia al número de veces que se ha exportado ese modelo. Una vez cargadas las variables del modelo se procede a generar el archivo Simulink en donde se encuentra el bloque que simula la dinámica de la planta (dedo), esto se hace escribiendo

en la ventana de comandos Adams_sys; para cualquier modelo, este comando no cambia y es quien genera el archivo donde se encuentra:

- La S-Function, quien representa el modelo ADAMS no lineal.
- El bloque adams_sub, contiene la S-Function, pero también crea varias variables útiles de MATLAB,
- El bloque State-Space representa un modelo de ADAMS linealizado.

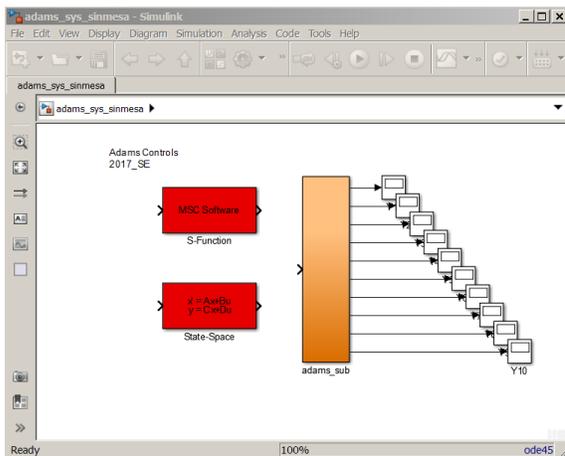


Figura 26. Modelo Adams_sys

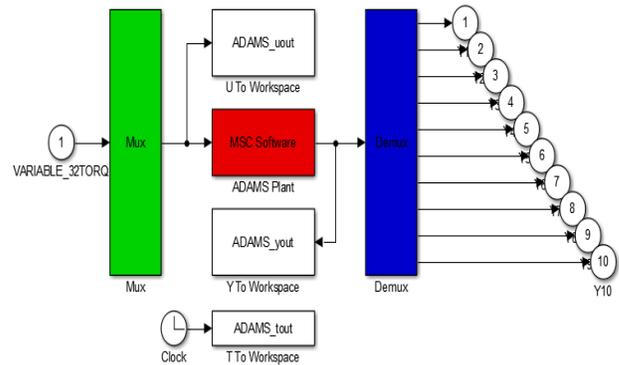


Figura 27. Bloque adams_sub

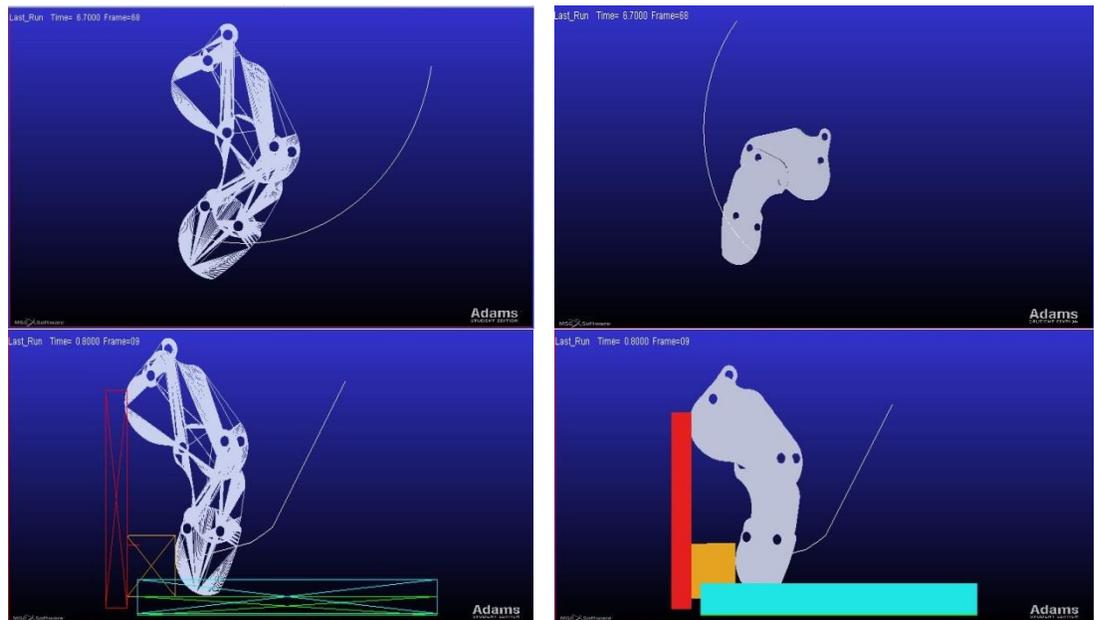


Figura 28. Modelo bajo la acción de la fuerza de gravedad sin objeto en el camino y con objeto en el camino

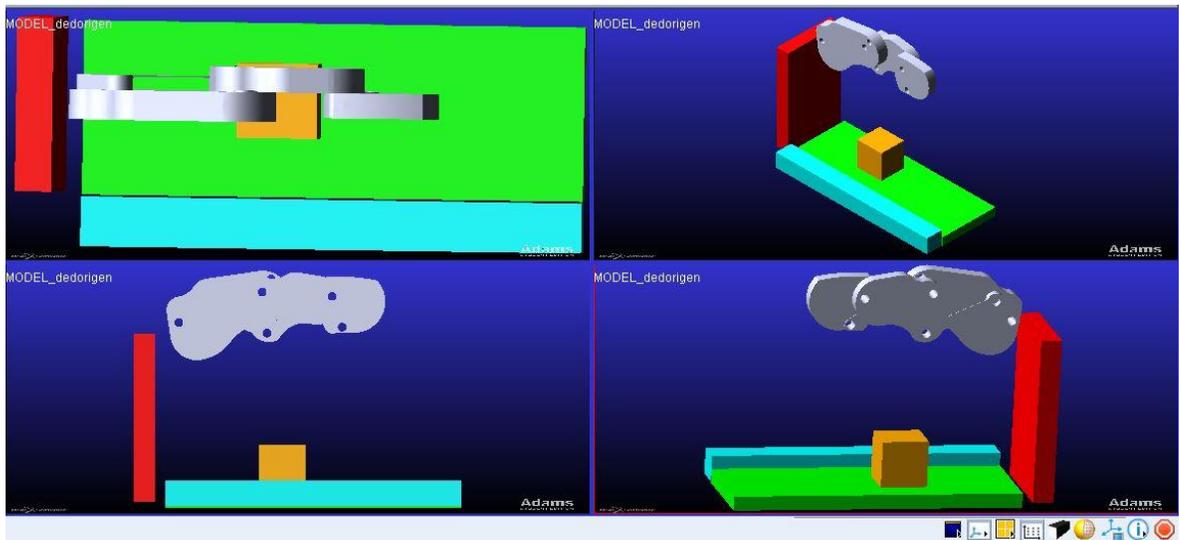


Figura 29. Vista renderizada de un ambiente simulado, con una mesa y un objeto

En el anexo A se encuentran todos los pasos desarrollados en Adams para la creación y exportación de nuestro modelo dinámico para el dedo sub-actuado.

Para la jacobiana, obtenida de manera simbólica en Matlab sección 1.7 realizamos el siguiente experimento a fin de obtener resultados comparativos y respaldar su validez Figura 30.

- Generación de una trayectoria polinómica de grado 5, idéntica a la usada en la sección 1.7.
- Exportación del modelo del dedo en Adams descritos en los pasos de 2.1, dejando como salida las variables de posición cartesiana x e y , la variable actuada θ_1 y cambiando la entrada del sistema por un movimiento rotacional que permite generar el movimiento deseado impuesto por la trayectoria grado 5.
- Se derivan individualmente las posiciones x e y a fin de obtener sus respectivas velocidades cartesianas para el análisis comparativo.

Este pequeño experimento nos permite respaldar la validez de la jacobiana obtenida en 1.7, comparando las velocidades cartesianas x e y en ambos experimentos. Al observar la Figura 24 obtenida por MATLAB y la Figura 31 obtenida por Adams, podemos decir que son muy similares, en cuanto a forma y a resultados numéricos; pero se aclara que en MATLAB se trabajó sin tener en cuenta condiciones físicas como gravedad, propiedades másicas, fricciones, momentos de inercia, fuerzas de contacto etc.

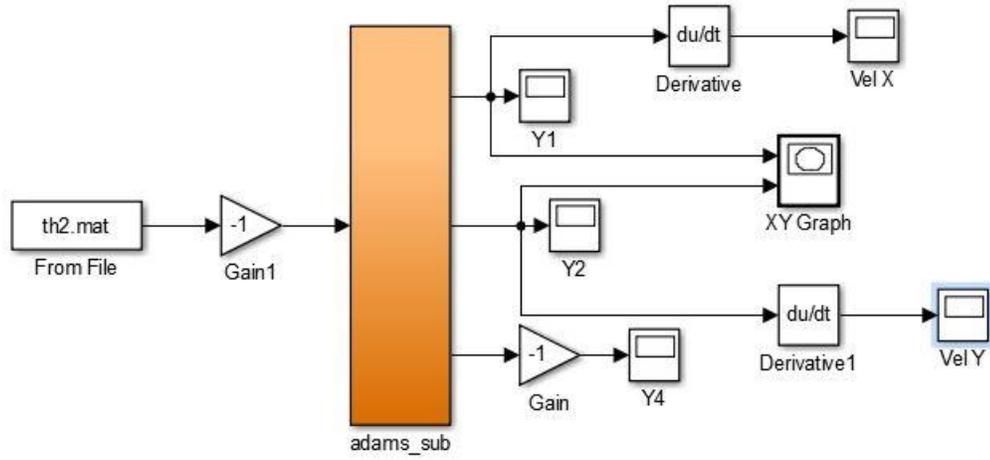


Figura 30. Simulink modelo del dedo en Adams

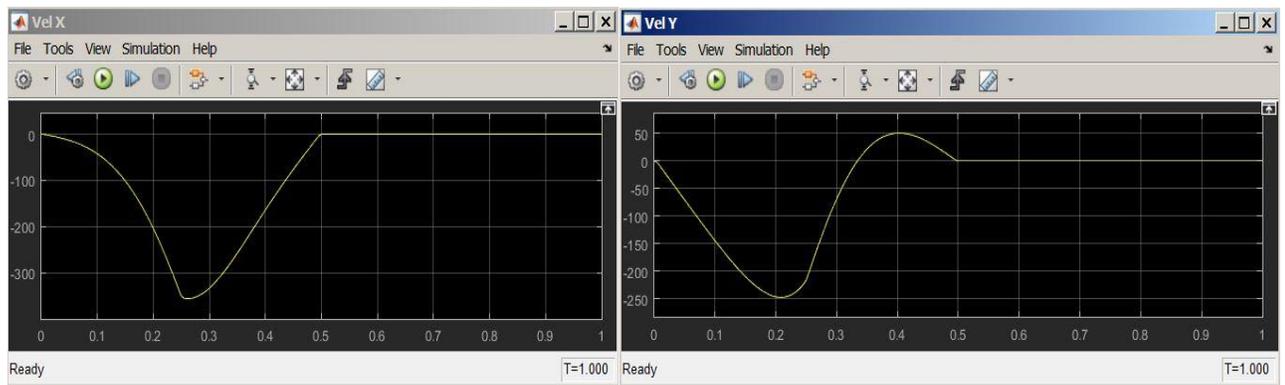


Figura 31. Jacobiana obtenida en simulación con Adams

3 HIPÓTESIS VIRTUAL SPRING DAMPER

La hipótesis virtual spring damper tiene dos importantes enfoques, el primero consiste en utilizar este método para determinar el modelo dinámico directo de sistemas robóticos complejos con cadenas cinemáticas cerradas y el segundo para obtener un algoritmo de control con importantes características y ventajas sobre otros controladores ampliamente usados. A continuación, se presenta cada uno de estos enfoques.

3.1 Enfoque virtual spring damper para el modelado y simulación de sistemas robóticos con cadenas cinemáticas cerradas

Uno de los principales enfoques de la hipótesis virtual spring damper se encuentra en la simulación dinámica de sistemas robóticos o mecánicos con cadenas cinemáticas cerradas, esta simulación permite el diseño de sistemas complejos, de entornos virtuales para el entrenamiento de operadores, de pantallas predictivas para teleoperación y permite el desarrollo de sistemas avanzados de control robótico [10].

La simulación de manipuladores robóticos serie ha sido un tema ampliamente abordado, por el contrario, la simulación de sistemas robóticos con cadenas cinemáticas cerradas ha sido un tema al que se le ha prestado menos atención. La simulación dinámica se realiza generalmente en dos pasos, el primero consiste en la generación del modelo dinámico y el segundo en la solución del modelo. La generación de los modelos dinámicos de manipuladores rígidos o flexibles con cadenas cinemáticas cerradas usando el principio de trabajo virtual para el cálculo de las dinámicas inversas se cree que es más eficiente que usar otras formulaciones multicuerpo. Sin embargo, cuando se utiliza para la solución de las dinámicas directas o solución dinámica, la solución de los modelos generados mediante este enfoque no es suficiente porque hay coordenadas generalizadas dependientes en las ecuaciones. Dado que las ecuaciones dinámicas de estos manipuladores robóticos son altamente no lineales y acopladas, la solución analítica de tales ecuaciones se piensa que es imposible y por lo tanto, deben utilizarse los enfoques numéricos. Además, la solución de la dinámica directa o simulación dinámica de manipuladores robóticos cerrados es más difícil que la de los robots serie dado que las restricciones cerradas causan coordenadas dependientes que aparecen en las ecuaciones dinámicas y el sistema es

indeterminado. Usualmente, para obtener un sistema determinado, las ecuaciones adicionales se obtienen diferenciando las restricciones. Sin embargo, la solución del conjunto de ecuaciones dinámicas aun no es suficiente debido a que las restricciones cinemáticas en desplazamientos y velocidades deben cumplirse al mismo tiempo. Existen tres métodos principales para incluir las restricciones cinemáticas de mecanismos robóticos complejos, el primero es el enfoque algebraico diferencial, el segundo es el enfoque de solución de cinemática en forma cerrada y el tercero es la familia de formulaciones denominada "acoplamiento de fuerza". Sin embargo, la mayoría de estos enfoques conducirán a formulaciones complejas que están acopladas y por lo tanto no pueden ser utilizados directamente para el desarrollo de algoritmos paralelos para la simulación dinámica en tiempo real utilizando multi-procesadores. En efecto, las formulaciones basadas en soluciones de forma cerrada consisten en ensamblar un sistema global de ecuaciones en el que todos los grados de libertad están acoplados mientras las formulaciones basadas en los multiplicadores de Lagrange también requieren la solución de un sistema lineal global con el objetivo de solucionar las fuerzas en las articulaciones con restricción.

Por los motivos anteriores se analiza el uso de la hipótesis virtual spring para generar los modelos dinámicos de manipuladores con cadenas cinemáticas cerradas. La idea básica de este enfoque ha sido utilizada en el análisis de vibración y en el modelado de sistemas mecánicos con restricciones. El enfoque virtual spring está basado en una formulación Lagrangiana y virtual springs, lo que conduce a un conjunto de ecuaciones dinámicas que pueden ser utilizadas para el análisis de la dinámica directa y para el de la inversa. Cuando se utiliza para el análisis de la dinámica inversa, se obtienen el mismo conjunto de ecuaciones de movimiento que se alcanzan utilizando el enfoque multiplicador de Lagrange o el principio de trabajo virtual. Cuando se utiliza para la simulación o análisis de la dinámica directa, las ecuaciones dinámicas obtenidas mediante este enfoque están completamente desacopladas y automáticamente incluyen las restricciones cinemáticas, lo que es ideal para desarrollar algoritmos de simulación paralelos de alto nivel para la simulación en tiempo real de estos sistemas robóticos.

3.2 Comparación entre el algoritmo de control virtual spring damper y las estrategias de control más utilizadas

Existen varias estrategias que pueden utilizarse para controlar un robot, estas pueden clasificarse de la siguiente manera [11]:

- Controladores que no se basan en el modelo matemático del robot.
- Controladores que utilizan el modelo matemático del robot.

Entre los controladores más utilizados se encuentra el control PID y el control por par calculado CTC, estas estrategias se van a abarcar con el objetivo de evidenciar los beneficios que presenta el control VSD frente a estas técnicas de control.

3.2.1 Control PID

Esta estrategia de control considera el mecanismo como un sistema lineal. Una importante característica es que no requiere el modelo matemático del robot o modelo dinámico inverso para calcular el torque requerido para cada una de las articulaciones actuadas. El control es descentralizado, de manera que se define un controlador PID con ganancias constantes para cada articulación motorizada del robot, esta estrategia de control es de fácil implementación y bajo costo, sin embargo, se pueden encontrar malas precisiones y desplazamientos excesivos en el caso de movimientos rápidos.

Según el espacio en el que se encuentre definida la consigna se tienen los siguientes esquemas para el PID.

3.2.1.1 PID Articular

El esquema de este controlador se muestra en Figura 26.

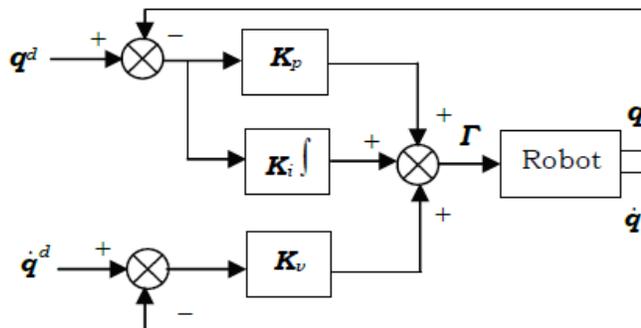


Figura 32. Esquema general control PID (Tomado de [10])

El bloque “Robot” contiene el modelo dinámico directo del robot necesario para realizar la simulación en un software como Simulink. La ley de control se define mediante la ecuación 44.

$$\gamma = Kp(q^d - q) + Kv(\dot{q}^d - \dot{q}) + Ki \int (q^d - q) dt \quad (44)$$

Como se puede observar tanto en el esquema general como en la ecuación de la ley de control, esta estrategia corrige el error tanto de posición como de velocidad en el espacio articular.

La posición, la velocidad y la aceleración deseada, las cuales son funciones del tiempo se conocen como consignas. Una consigna o señal de referencia puede estar definida en el espacio articular con el objetivo de realizar pruebas en el robot o para generar movimientos simples, y también puede definirse en el espacio operacional o espacio de trabajo del robot con la finalidad de seguir los requerimientos establecidos.

Si se desea seguir una trayectoria para cada articulación, es decir, si existe una consigna para cada una de las articulaciones actuadas del robot, estas ingresan directamente como entradas del controlador. Pero si la consigna es cartesiana, es decir, si se ha definido una por cada eje del espacio operacional del robot, estas referencias deben ser transformadas en consignas articulares, debido a que como se ha observado, la ley de este esquema de control PID permite el seguimiento de trayectorias en el espacio articular.

El modelo que permite obtener todas las posibles soluciones de las posiciones articulares que permiten situar al órgano terminal en una posición cartesiana deseada es el modelo geométrico inverso MGI. Sin embargo, el inconveniente de este modelo radica en que como se ha mencionado para alcanzar una posición deseada en el espacio de trabajo del robot, existen diversas posibilidades de posiciones articulares y por tanto debe elegirse una de estas.

Por otra parte, es necesario tener en cuenta que esta estrategia de control trabaja como un sistema bastante acoplado, es decir, que al sintonizar las ganancias del controlador que van a permitir seguir la trayectoria deseada de una de las articulaciones, esto influye en las ganancias ya sintonizadas para cada una de las articulaciones actuadas del robot.

3.2.1.2 PID Cartesiano

Para este esquema de control en el espacio operacional, el lazo de control se obtiene reemplazando q por x en la ecuación 44 y transformando la señal de control en el espacio de trabajo del robot, al espacio articular mediante la matriz Jacobiana, de la siguiente manera.

$$\tau = J^T [K_P(x_d - x) + K_d(\dot{x}_d - \dot{x}) + K_i \int (x_d - x)] \quad (45)$$

3.2.2 Control por par calculado (CTC)

Esta estrategia de control permite evoluciones rápidas del robot y una gran precisión, se conoce como control por par calculado, control por desacoplamiento no lineal o control dinámico. Este controlador está basado en el modelo matemático del robot, es decir, que exige el cálculo del modelo dinámico inverso en línea y por tanto requiere conocer con cierta exactitud los valores numéricos de los parámetros dinámicos.

Según el espacio en el que se encuentre definida la consigna, se tienen los siguientes esquemas para el CTC.

3.2.2.1 CTC Articular

Esta estrategia de control permite desacoplar las articulaciones, de manera que la sintonización de las ganancias para cada una de las articulaciones es independiente, a diferencia del control PID, en el que además deben sintonizarse tres ganancias para cada articulación en lugar de dos como sucede en el CTC.

3.2.2.2 CTC Operacional. Si la consigna está definida en el espacio de trabajo del robot, y este tiene varias articulaciones actuadas, las coordenadas cartesianas se deben convertir en posiciones articulares mediante el modelo geométrico inverso, que como ya se ha mencionado presenta el inconveniente de que para alcanzar una posición deseada en el espacio operacional, que es aquel en el que se ubica el órgano terminal, existen varias posiciones de las articulaciones que lo permiten, luego de realizar esto se aplica el control por par calculado en el espacio articular visto anteriormente.

Sin embargo, a diferencia del control PID en el que si se tiene una consigna cartesiana se debe necesariamente utilizar el MGI, el CTC permite establecer un esquema que permite trabajar directamente con este tipo de consigna, para ello se definen las ecuaciones dinámicas del robot en el espacio operacional.

El control CTC operacional a diferencia de las estrategias de control presentadas, no requiere implementar el modelo geométrico inverso, sin embargo, si requiere calcular la matriz Jacobiana y su inversa.

Por otra parte, al realizar un control en el espacio de trabajo, se sintonizan las ganancias por cada coordenada cartesiana y no por cada articulación como sucede en el control PID y CTC articular, esta es una ventaja cuando el robot tiene más articulaciones actuadas que dimensiones en el espacio de trabajo.

Teniendo en cuenta los controladores PID y CTC, se presenta una estrategia de control que se va a utilizar para realizar el control de posición de la yema de un dedo índice que corresponde a un mecanismo sub-actuado. De esta manera, es posible entender las ventajas que presenta este innovador método de control con respecto a las estrategias mencionadas.

3.2.3 Algoritmo de control basado en la hipótesis virtual spring damper.

Con el objetivo de solucionar el problema de los grados de libertad según el neurofisiólogo soviético Nikolai Bernstein se ha propuesto este método de control natural. El problema consiste en encontrar las señales de control que se deben ejercer a las articulaciones actuadas y que conducen al movimiento del órgano terminal hacia una posición deseada en el plano xy , debido a que surgen varias inversas qd (posiciones articulares deseadas) que permiten alcanzar a xd (posiciones cartesianas deseadas), de manera que obtener las cinemáticas inversas desde el espacio operacional $x \in \mathbb{R}^2$ al espacio articular se convierte en una tarea que incrementa el costo computacional [12].

Como se mencionó anteriormente, para un robot con varias articulaciones actuadas y un esquema de control con consigna cartesiana, el PID y el CTC articular deben utilizar el modelo geométrico inverso, el cual tiene el inconveniente planteado por Bernstein. Para resolverlo, se propuso que si una posición deseada está dada por medio de determinar su velocidad, es posible calcular la velocidad de las articulaciones mediante la inversa de la matriz Jacobiana, debido a que esta relaciona las velocidades en el espacio operacional con respecto a las velocidades articulares. Una vez, conocidas las velocidades articulares deseadas, es posible mediante su derivada, obtener las aceleraciones articulares, las cuales se reemplazan en el modelo dinámico inverso en el que se basa el control por par

calculado, de manera que quedan definidas las ecuaciones matemáticas del robot en el espacio operacional y así es posible obtener la estrategia del CTC operacional, el cual no requiere del modelo geométrico inverso, dado que realiza la corrección del error de posición, velocidad y aceleración, según se desee, en el espacio de trabajo del robot. Sin embargo, aunque el CTC operacional no utiliza el modelo geométrico inverso, si requiere del cálculo de la inversa de la matriz Jacobiana, además del modelo dinámico inverso.

Debido a los inconvenientes y complejidad de las estrategias de control mencionadas, se propuso un método de control natural [13, 14, 15] que ha evidenciado su efectividad en robots con grados de libertad redundantes, como el de la Figura 27.

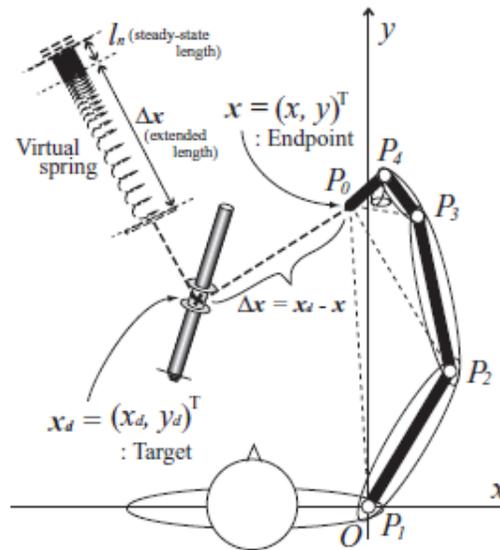


Figura 33. Esquema general control PID (Tomado de [10])

La ley de control de esta estrategia está definida por la ecuación 46.

$$u = -C\dot{q} - J^T(q)k\Delta x \quad (46)$$

Donde C denota una matriz definida positiva en la diagonal como sigue,

$$C = \text{diag}(c_1, \dots, c_n) \quad (47)$$

Esta estrategia es más simple que la del CTC operacional, y una vez los factores de amortiguamiento C y el único parámetro de rigidez k , se seleccionen cuidadosamente, se generan movimientos de alcance suaves por medio de una trayectoria del órgano terminal casi recta, perfiles de velocidad en forma de campana y perfiles de aceleración de doble pico típicamente vistos en el caso de los movimientos multi-articulares de la habilidad humana.

Este algoritmo de control puede deshacerse de cualquier movimiento inherente a los sistemas redundantes, además de ello, solo expone pequeñas fluctuaciones en la trayectoria del órgano terminal, sin embargo, los torques calculados bajo incertidumbre en los parámetros de los enlaces del mecanismo inducen fluctuaciones notables en esta trayectoria en el espacio operacional. Tales fluctuaciones son más notorias en los casos de robot con grados de libertad redundantes.

Una importante característica de esta ley de control es que no requiere definir ninguna trayectoria deseada para el órgano terminal en un espacio operacional planar. Por otra parte, una desventaja que presenta es que la elección de los factores de amortiguamiento no se ajusta a la escala de coeficientes de viscosidad de los músculos humanos.

Este método de control que es conocido como la hipótesis virtual spring ha sido extendido a un alcance bidimensional, basado en una hipótesis Virtual Spring Damper y a un alcance en las tres dimensiones bajo el efecto de la gravedad. En ambos casos, los movimientos se realizan usando una señal de control simple similar a la de la Ecuación 45. Seleccionando los parámetros del controlador cuidadosamente, las fuerzas como las viscosas en las articulaciones trabajan armoniosamente mediante el esfuerzo de la fuerza del resorte.

La hipótesis Virtual Spring Damper para un alcance bidimensional está basada en la definición de un resorte y un amortiguador en paralelo como se ilustra en la Figura 28.

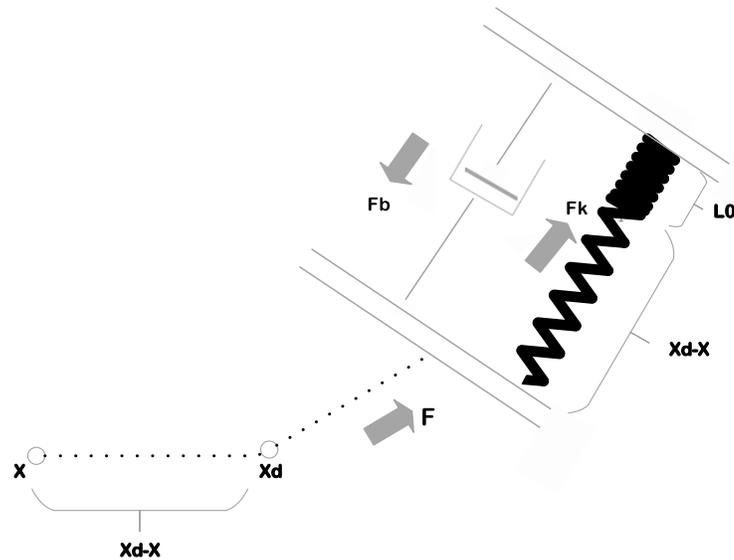


Figura 34. Resorte y amortiguador en paralelo

Por la segunda ley de Newton se tiene que,

$$F = m\ddot{x} \quad (48)$$

Donde F es la fuerza resultante aplicada sobre el mecanismo, la cual corresponde a la suma de las fuerzas que actúan sobre él, de manera que,

$$F = F_K - F_B \quad (49)$$

Donde F_K es la fuerza ejercida por el resorte, la cual, por la Ley de Hooke, y teniendo en cuenta la Figura 28, se define en la ecuación 50, esta fuerza virtual genera la ilusión de una fuerza que atrae el efector final, que corresponde a la yema de dedo en este proyecto, desde su posición actual hasta su posición deseada. Por su parte F_B , corresponde a la fuerza de amortiguamiento, definida en la ecuación 51, esta fuerza del amortiguador virtual se opone al movimiento, de manera que el mecanismo sub-actuado propuesto va perdiendo velocidad, dado que F_B es proporcional a su velocidad.

$$F_K = -k(x - x_d) \quad (50)$$

$$F_B = b \frac{dx}{dt} \quad (51)$$

Reemplazando las ecuaciones 50 y 51 en la ecuación 49, se tiene la fuerza neta sobre el mecanismo, definida de la siguiente manera,

$$F = -k(x - x_d) - b\dot{x} \quad (52)$$

Despejando la aceleración del cuerpo, \ddot{x} , de la ecuación 48, se tiene,

$$\ddot{x} = -\frac{k}{m}(x - x_d) - \frac{b}{\sqrt{m}} \frac{\sqrt{k}}{\sqrt{km}} \dot{x} \quad (53)$$

Se realizan las siguientes definiciones,

$$w_n^2 = \frac{k}{m} \quad (54)$$

$$\zeta = \frac{b}{2\sqrt{km}} \quad (55)$$

Sustituyendo w_n y ζ en la ecuación 53,

$$\ddot{x} = -w_n^2(x - x_d) - 2\zeta w_n \dot{x} \quad (56)$$

Aplicando la Transformada de Laplace a la ecuación anterior, se obtiene

$$\frac{x(s)}{x_d(s)} = \frac{w_n^2}{s^2 + 2\zeta w_n s + w_n^2} \quad (57)$$

La ecuación anterior garantiza que x sigue a x_d con frecuencia natural w_n y factor de amortiguamiento ζ , factores que dependen del coeficiente de rigidez del resorte k y del coeficiente de amortiguamiento c .

Sustituyendo la ecuación 56 en la ecuación 48, se tiene la siguiente ecuación para la fuerza resultante F en términos de w_n y ζ ,

$$\begin{aligned} F &= m\ddot{x} \\ &= -(w_n\sqrt{m})^2(x - x_d) - (2\sqrt{m}\zeta)(\sqrt{m}w_n)\dot{x} \end{aligned} \quad (58)$$

Se realizan las siguientes definiciones,

$$\hat{\zeta} = 2\sqrt{m}\zeta \quad (59)$$

Despejando k de la ecuación 54,

$$k = w_n^2 m \quad (60)$$

Sustituyendo las ecuaciones 59 y 60 en la ecuación 58, se tiene,

$$F = -k(x - x_d) - \hat{\zeta}\sqrt{k}\dot{x} \quad (61)$$

En la ecuación anterior, k y $\hat{\zeta}$, corresponden a parámetros que deben identificarse.

Mediante la matriz Jacobiana del manipulador, es posible relacionar F con el torque virtual aplicado a la articulación activa τ_{sd} , de la siguiente manera,

$$\tau_{sd} = Jma^T F \quad (62)$$

Además, esta hipótesis virtual, considera amortiguadores en las articulaciones activas del mecanismo robótico por lo que la señal de control es,

$$\tau_v = -C\dot{q} + Jma^T F \quad (63)$$

A continuación, se realiza un análisis del equivalente de la segunda ley de Newton para la rotación, definida por la siguiente ecuación,

$$M = I\ddot{q} \quad (64)$$

Donde M es el momento o torque neto aplicado al cuerpo, el cual es la suma de los torques que actúan sobre él. De esta manera, para el mecanismo propuesto se tiene,

$$M = -\tau_a + \tau_v - c_f\dot{q} - g' \quad (65)$$

Donde,

τ_a : Torque aplicado a la articulación activa

τ_v : Torque virtual

$c_f\dot{q}$: Torque debido al amortiguamiento por fricción seca

g' : Torque debido a la gravedad

Reemplazando la ecuación 65 en la ecuación 64,

$$-\tau_a + \tau_v - c_f\dot{q} - g' = I\ddot{q} \quad (66)$$

Si elige un par aplicado, tal que,

$$\tau_a = \tau_v - c_f\dot{q} - g' \quad (67)$$

Entonces,

$$\ddot{q} = 0 \quad (68)$$

Es decir, que, con el par aplicado, el sistema vuelve a su estado de reposo inicial.

Según el modelo dinámico inverso, el par aplicado a la articulación activa está dado por la siguiente ecuación,

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau_a \quad (69)$$

Reemplazando el par aplicado hallado en la ecuación 67, en la ecuación anterior, se tiene,

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau_v - c_f\dot{q} - g' \quad (70)$$

Despejando el par virtual,

$$M(q)\ddot{q} + (C(q, \dot{q}) + c_f)\dot{q} + g(q) + g' = \tau_v \quad (71)$$

Linealizando la ecuación anterior y aplicando Transformada de Laplace, se tiene,

$$\frac{q(s)}{\tau_v(s)} = \frac{K_r W_{nr}^2}{s^2 + 2\zeta_r W_{nr} s + W_{nr}^2} \quad (72)$$

Aplicando la Transformada de Laplace al modelo cinemático directo, se tiene,

$$x(s) = \bar{J}q(s) \quad (73)$$

Reemplazando la ecuación anterior en la ecuación 72,

$$\frac{x(s)}{\tau_v(s)} = \frac{K_r W_{nr}^2 \bar{J}}{s^2 + 2\zeta_r W_{nr} s + W_{nr}^2} \quad (74)$$

Definiendo,

$$G_r(s) = \frac{K_r W_{nr}^2 \bar{J}}{s^2 + 2\zeta_r W_{nr} s + W_{nr}^2} \quad (75)$$

Reemplazando la ecuación anterior en la ecuación 74,

$$x(s) = G_r(s) \tau_v(s) \quad (76)$$

Del modelo virtual, se tiene,

$$x(s) = G_v(s) x_d(s) \quad (77)$$

Reemplazando la ecuación anterior en la ecuación 76,

$$\tau_v(s) = \frac{G_v(s)}{G_r(s)} x_d(s) \quad (78)$$

De esta manera, si se ajustan adecuadamente los parámetros del modelo virtual, la posición de la yema del dedo va a alcanzar la posición deseada.

Así, este método de control para movimientos planares sin efecto de la gravedad es expresado por la ecuación 79. Esta señal representa un par paralelo de amortiguador mecánico y resorte que atrae el órgano terminal a la posición deseada como se observa en la Figura 29 para un brazo robótico. Además, considera amortiguadores virtuales en cada articulación activa del robot.

$$\begin{aligned} u &= -C_0 \dot{q} - J^T(q) (\zeta \sqrt{k} \dot{x} + k \Delta x) \\ &= -C_0 \dot{q} - \zeta \sqrt{k} J^T(q) J(q) \dot{q} - J^T k \Delta x \end{aligned} \quad (79)$$

Donde C_0 se escoge como sigue

$$C_0 = \zeta_0 \text{diag}(c_1, \dots, c_n) \quad (80)$$

Con constante positiva ζ_0 entre 0 y 1.

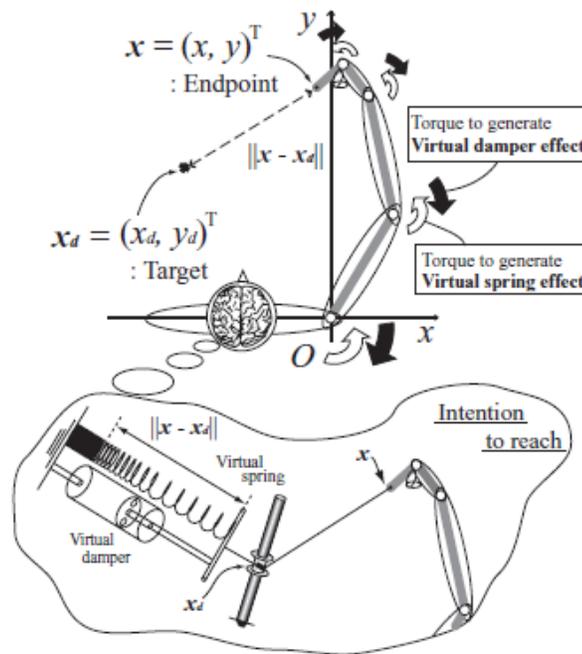


Figura 35. Hipótesis virtual spring damper aplicada a un brazo robótico (Tomada de [12])

Para conocer el sentido físico de este algoritmo de control, el término $-kJ^T(q)\Delta x$ se explica desde la hipótesis virtual spring. Este término es una realimentación del espacio operacional que se basa en la medida de la posición del órgano terminal x mediante un sensor. En el caso del alcance humano, los torques ejercidos $-kJ^T(q)\Delta x$ deben ser generados desde un grupo de músculos que están dotados con una energía potencial total generada por la excitación de las señales neuro motoras desde el sistema nervioso central. Este hecho puede ser bien interpretado mediante la suposición de un resorte virtual, trazando el punto final del órgano terminal al punto deseado con la fuerza equivalente al vector $k(x_d - x)$ que es equivalente a $-k\Delta x$. En este modelo hipotético la i -ésima articulación es actuada por medio del torque $-kJ^{Ti}(q)\Delta x$ y también sujeta a la fuerza de amortiguamiento $-c_i\dot{q}_i$.

Por otra parte, hay una falta de documentos aun en revistas de fisiología que detallen las funciones efectivas de los torques tipo viscoso $-c_i\dot{q}_i$ en la convergencia sinérgica de la trayectoria del órgano terminal a la posición deseada.

La señal de control genera efectivamente movimientos de varias articulaciones suaves y naturales. Cada factor de amortiguamiento pasivo $\zeta_0 c_i$ correspondiente a cada articulación puede ser reducido al rango del 20% de c_i .

De acuerdo a los hallazgos fisiológicos, el término $-\zeta\sqrt{k}J^T(q)\dot{x}$ visto en la ecuación 58 refleja el torque de amortiguamiento generado por las contracciones de los músculos agonistas y antagonistas. La suposición de tal combinación de amortiguador mecánico virtual y el resorte que llevaría el órgano terminal a la posición deseada da lugar a la generación de señales de control con “impedancia” regulada.

El enfoque de control de la hipótesis virtual spring damper puede extenderse a movimientos de alcance tridimensional bajo el efecto de la fuerza de gravedad. El término de gravedad que viene de la energía potencial $P(q)$ se representa en la ecuación 81.

$$g(q) = \left(\frac{dP}{dq}\right)^T \quad (81)$$

Entonces, se adiciona un estimador del término anterior al algoritmo de control de la ecuación 79, de manera que se obtiene la siguiente señal para controlar movimientos bajo el efecto la gravedad,

$$u = -C_0\dot{q} - J^T(q)(\zeta\sqrt{k}\dot{x} + k\Delta x) + Y\hat{\theta} \quad (82)$$

Donde Y representa un regresor que para el mecanismo sub-actuado propuesto corresponde a,

$$Y = g \quad (83)$$

Por su parte, $\hat{\theta}$ representa los estimados de los términos de la gravedad. Para el mecanismo sub-actuado, $\hat{\theta}$ es un único estimado de un valor real que corresponde a,

$$\theta = \frac{1}{g}(g1(q)) \quad (84)$$

El estimado $\hat{\theta}$ se calcula mediante la siguiente fórmula causal,

$$\hat{\theta}(t) = \hat{\theta}(0) - \int_0^t \gamma^{-1}Y(q(\tau))s(\tau)d\tau \quad (85)$$

$$s(t) = \dot{q}(t) + C^{-1}kJ^T(q(t))\Delta x(t) \quad (86)$$

Donde γ corresponde a un valor que debe ser positivo.

Así, este método es un esquema de control de posición que mediante la simulación de componentes virtuales permite generar el torque que debe aplicarse a las articulaciones con el objetivo de seguir una trayectoria deseada. Estas articulaciones producen el mismo efecto que los elementos virtuales, creando así la ilusión de que estos se encuentran conectados a la estructura real del robot [16].

Como se presentó, este método de control natural trabaja directamente en el espacio operacional a diferencia del control PID y el control CTC, los cuales solo permiten corregir el error en el espacio articular, por lo que, si se define una trayectoria cartesiana, se requiere necesariamente la implementación del modelo geométrico inverso MGI, el cual presenta el inconveniente ya mencionado de no generar una solución única, de manera que de todas las posibles soluciones debe escogerse la apropiada, además, el VSD además de corregir el error en el espacio operacional, no requiere necesariamente definir una trayectoria cartesiana, si se desea solo se define el punto deseado y como se mencionaba, el algoritmo permite alcanzar esta posición siguiendo una trayectoria casi recta, similar a los movimientos humanos. Además, esta estrategia de control no está basada en el modelo matemático del robot, es decir, que no necesita del cálculo en línea del modelo dinámico inverso para la generación del torque aplicado a los actuadores, a diferencia de otros controladores como el control robusto, predictivo, adaptativo, por par calculado y entre otros que se encuentran basados en este modelo. De esta manera, al no ser necesario calcular estos modelos, la ley de control se simplifica y se minimizan los recursos computacionales requeridos.

Para analizar una gran ventaja de este algoritmo de control en el agarre de objetos se presenta la siguiente aplicación.

3.2.3.1 Agarre de una prótesis de mano utilizando el algoritmo de control virtual spring damper

El agarre puede ser clasificado en precisión y en poder. En la precisión de agarre, el objeto es sostenido por la punta de los dedos, mientras que, en el poder de agarre, todo el dedo está en contacto con el objeto.

La mayoría de los métodos requieren el conocimiento de la forma del objeto, esta información visual no está disponible directamente para el controlador de la prótesis de mano, la única información que se tiene es la de la señal electromiográfica EMG relacionada a las actividades musculares del brazo del paciente. Sin embargo, normalmente esta señal no se encuentra para todas las articulaciones individuales. Por otra parte, el uso de la señal EMG para controlar todos los movimientos requiere demasiada atención durante el agarre, lo que conduce a la fatiga del paciente. Por tal motivo, se requiere que una parte de la información de comando sea proporcionada por la señal EMG y el resto sea dada por el controlador. Para ello, se ha seleccionado la estrategia de control virtual spring damper, debido a que este algoritmo permite abordar de manera eficiente el problema tanto de precisión como de poder de agarre sin el conocimiento de la forma del objeto.

3.2.3.1.1 Precisión en el agarre

La ley de control VSD permite obtener un agarre preciso sin que sea necesario conocer la posición y la forma del objeto que se desea agarrar al definir un resorte y un amortiguador virtual entre las yemas de los dedos que van a sujetar el objeto, así los dedos se van a atraer entre sí y van a agarrar el objeto que se encuentra en el medio. En este caso, el paciente debe colocar la mano cerca al objeto y en una posición apropiada [16].

Los altos valores de K que corresponde a la constante de rigidez del resorte virtual, generan movimientos más rápidos de los dedos, así como, mayor fuerza de agarre, Así al definir K proporcional a la señal EMG, el paciente tiene control sobre la velocidad del movimiento y la fuerza de agarre. Los valores apropiados de los coeficientes de amortiguamiento de las articulaciones de los dedos se evalúan mediante ensayo y error para alcanzar posiciones cercanas a la mano normal, de esta manera se eligen los más adecuados.

3.2.3.1.2 Poder en el agarre

El algoritmo de control virtual spring damper permite manejar eficientemente el problema del agarre sin el conocimiento de la forma del objeto. En esta aplicación se define una trayectoria deseada en forma de semicírculo con el objetivo de tener un total agarre del objeto, la variable tiempo de la trayectoria definida es proporcional a la señal EMG que se escala para cambiar entre 0 y π , luego de pasar este semicírculo, la punta del dedo va hacia el centro para realizar un agarre

más comprimido, lo que se logra definiendo un punto deseado cerca al centro. El objetivo no es seguir exactamente la trayectoria definida, debido a que si el objeto es grande, dado el contacto de la prótesis de la mano y el objeto, sería imposible seguir la trayectoria, la cual está definida con el objetivo de lograr el agarre. Esta estrategia de control se implementó para lograr el agarre total de tres diferentes objetos y efectivamente se logró el objetivo. De manera que se evidencia que independientemente de la forma del objeto, esta estrategia de control es eficiente [20].

Analizando estos importantes enfoques de la hipótesis virtual spring damper, se decide implementar el enfoque aplicado a la obtención de la estrategia de control, dado que, aunque es interesante obtener el modelo dinámico directo del mecanismo sub-actuado robótico con cadenas cinemáticas cerradas propuesto, este modelo puede obtenerse mediante diversos Software como Adams, que corresponde al entorno virtual utilizado en este proyecto. Por otro lado, al implementar el método de control, se obtienen las ventajas de trabajar en el espacio operacional dado un algoritmo simple que permite reducir costos y obtener muy buenos resultados como se observa en diversas aplicaciones [12, 16, 17, 18, 19]. De esta manera es posible analizar su desempeño y sus ventajas para el control de posición del sistema robótico propuesto.

4 OPTIMIZACIÓN PARA SINTONIZAR CONTROL VSD y PID

En la literatura estudiada acerca del algoritmo de control VSD, los parámetros del controlador se eligen basándose en los resultados de pruebas de desempeño con distintos valores, es decir, a modo de ensayo-error. En este proyecto se propone que esta selección se realice mediante un algoritmo de optimización, algoritmo con el que también se van a sintonizar los parámetros de un controlador PID con el objetivo de comparar el desempeño de ambos controladores que no se encuentran basados en el modelo matemático del robot y que tampoco requieren el modelo geométrico inverso MGI para el control del dedo robótico sub-actuado, debido a que como se mencionaba el método de control natural VSD ha sido diseñado para realizar corrección del error en el espacio operacional y por su parte, el control PID en este caso en particular que solo requiere controlar una articulación no necesita de este modelo.

La optimización se refiere a la minimización o maximización de funciones. Las técnicas de optimización permiten localizar un conjunto de parámetros de diseño o decisiones que ofrecen el mejor resultado posible. Para ello, es necesario definir la función objetivo, la cual calcula la cantidad deseada que se debe minimizar o maximizar, es posible agregar restricciones que limiten los valores posibles para los parámetros de diseño o las decisiones [21].

El toolbox de optimización de MATLAB consta de funciones que realizan minimización o maximización de funciones, generalmente no lineales. También provee funciones para resolver ecuaciones no lineales y para solucionar problemas de mínimos cuadrados y de ajuste de datos.

Para este proyecto se requiere calcular los parámetros del controlador VSD con el objetivo de minimizar el error entre la posición cartesiana deseada y la posición actual de la yema del dedo robótico sub-actuado, de manera que se logre alcanzar este punto definido en el espacio bidimensional de manera eficiente. Asimismo, se desean calcular las ganancias del controlador PID con el objetivo de seguir una trayectoria articular ya definida. Para ello, se ha seleccionado como un problema de mínimos cuadrados no lineales.

Una función disponible del toolbox para solucionar problemas de mínimos cuadrados no lineales es `lsqnonlin`. Esta rutina de optimización requiere la definición de un archivo M que contiene la función a minimizar, también se puede usar un objeto en línea creado desde una expresión de MATLAB. Las opciones de optimización pasadas a las rutinas cambian los parámetros de optimización. Los gradientes se calculan utilizando un método de diferencias finitas adaptativo a

menos que se suministren en una función. Hay algoritmos considerados de media escala y otros de gran escala que son diseñados para manejar los problemas a gran escala de manera eficiente.

El toolbox incluye dos algoritmos considerados de media escala para resolver problemas de mínimos cuadrados no lineales:

- El algoritmo trust-region reflective implementa un enfoque de región de confianza especializado para problemas de mínimos cuadrados no lineales. Se aplica a los problemas sin restricciones y los problemas con restricciones de límite.
- El algoritmo Levenberg-Marquardt implementa un método Levenberg-Marquardt estándar. Se usa para problemas sin restricciones.

La función `lsqnonlin` resuelve problemas de mínimos cuadrados no lineales de la forma:

$$\min_x \|f(x)\|_2^2 = \min_x (f_1(x)^2 + f_2(x)^2 + \dots + f_n(x)^2) \quad (87)$$

Con límites inferiores y superiores opcionales `lb` y `ub` en las componentes de `x`. `x`, `lb` y `ub` pueden ser vectores o matrices. En lugar de calcular el valor $\|f(x)\|_2^2$ (la suma de los cuadrados), `lsqnonlin` requiere la función definida por el usuario para calcular la función en forma vectorial

$$f(x) = \begin{bmatrix} f_1(x) \\ f_2(x) \\ \vdots \\ f_n(x) \end{bmatrix} \quad (88)$$

La función `lsqnonlin` permite varias sintaxis, de las cuales se ha seleccionado la siguiente:

$$x = \text{lsqnonlin}(\text{fun}, x_0, lb, ub, options) \quad (89)$$

Empieza en el punto x_0 y encuentra un mínimo de la suma de los cuadrados de las funciones descritas en fun. La función fun debería retornar un vector de valores y no la suma de los cuadrados de los valores. (El algoritmo calcula implícitamente la suma de los cuadrados de las componentes de fun (x)). Lb y ub corresponden a los límites inferiores y superiores para el diseño de las variables x de manera que la solución siempre está en el rango $lb \leq x \leq ub$. Si los límites de entrada especificados para un problema son inconsistentes, la salida x tomará el valor de x_0 , los componentes x_0 que no se encuentren dentro de los límites establecidos se colocan dentro de estos. Esta función se encarga de minimizar la función fun con las opciones de optimización especificadas en options.

A continuación, se presenta el procedimiento para sintonizar mediante optimización los parámetros del control PID y el control VSD.

4.1 Optimización para sintonizar el control PID

El esquema del control PID de posición de un dedo robótico sub-actuado se presenta en la Figura 36.

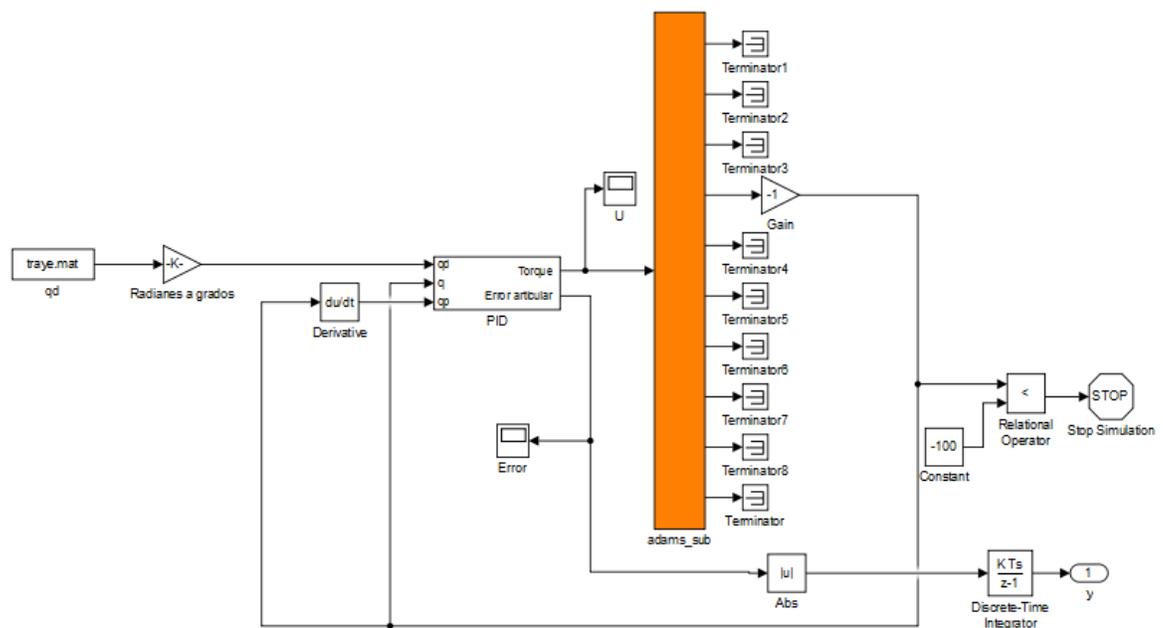


Figura 36. Esquema del control PID

Inicialmente, las ganancias de este algoritmo de control, se sintonizan de forma manual, bajo ensayo y error, con el objetivo de tener un conjunto de valores para fijar como punto inicial o alguno de los límites de la rutina de optimización. De manera que se obtienen los valores $K_p = 45$, $K_i = 0.5$, $K_d = 0.2$. En las siguientes Figuras se observa el comportamiento del sistema en lazo cerrado.

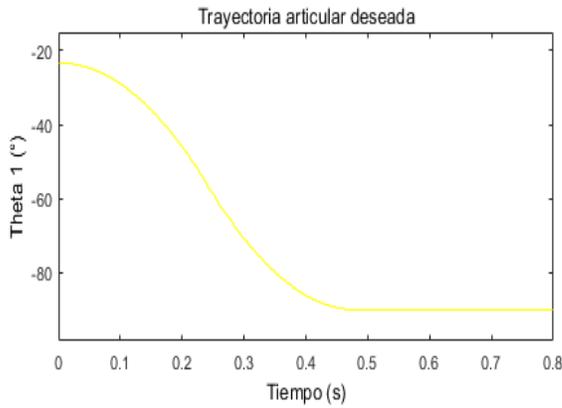


Figura 37. Trayectoria articular deseada

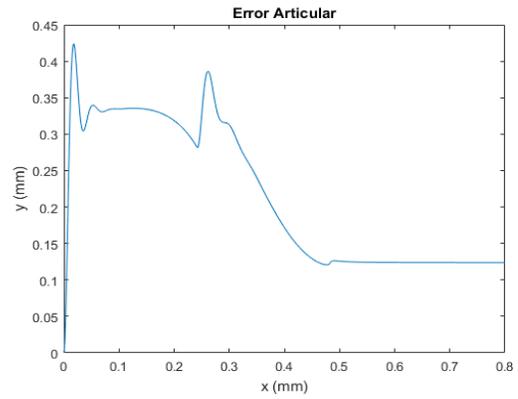


Figura 38. Error de seguimiento de la trayectoria articular

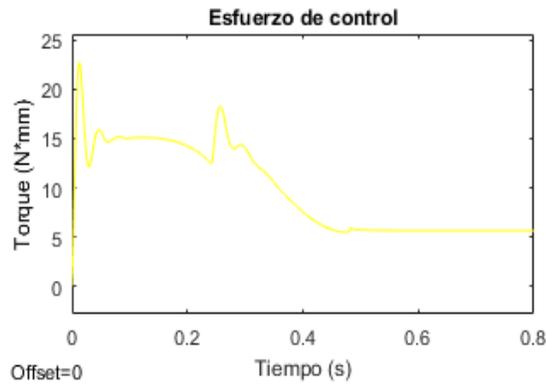


Figura 39. Esfuerzo de control

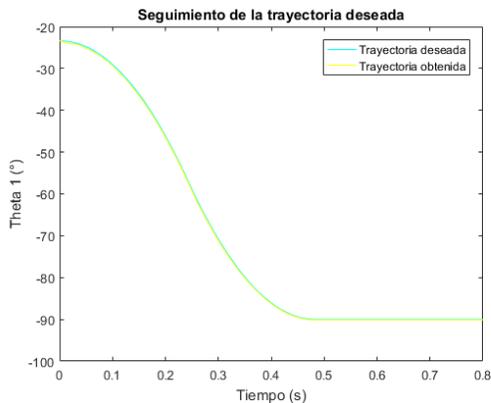


Figura 40. Seguimiento de la trayectoria articular deseada

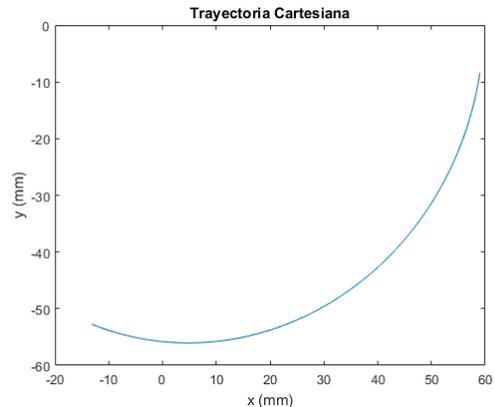


Figura 41. Trayectoria cartesiana obtenida

Como se observa en la Figura 38, la diferencia entre la trayectoria articular deseada y la obtenida es pequeña, hasta que finalmente debido al esfuerzo que realiza el controlador visto en la Figura 39, el error se aproxima a cero. El seguimiento de la trayectoria articular, genera la trayectoria cartesiana de la yema del dedo de la Figura 40.

Con estos parámetros del controlador obtenidos bajo ensayo y error, se tiene el punto inicial para realizar una sintonía mediante mínimos cuadrados no lineales. Debido a que se requiere minimizar la diferencia entre la trayectoria articular deseada y la actual para todos los pasos de tiempo de 0 a 1 se genera una función multiobjetivo, es decir, una función para cada paso de tiempo.

Lo primero que se debe hacer es definir una función en MATLAB, a la que se le ha llamado tracklsq, esta función recibe y define los parámetros utilizados en la simulación, la cual ejecuta mediante el comando sim, la función simset se utiliza para indicar a sim que ejecute la simulación en el espacio de trabajo actual. Al realizar la simulación, se calcula el error articular, esta variable se ha llevado a un bloque de puerto de salida en el diagrama de bloques de Simulink, de manera que queda definida en el espacio de trabajo actual correspondiente al espacio de trabajo de tracklsq, este vector de valores corresponde a la función multiobjetivo que se desea minimizar.

El segundo paso es llamar a la rutina de optimización, como se mencionaba anteriormente en la descripción de la sintaxis de la función lsqnonlin, deben definirse los parámetros iniciales para las ganancias del controlador alrededor de las cuales, la rutina de optimización va a calcular los parámetros que minimicen la suma de los cuadrados de las funciones descritas en tracklsq, estos valores

corresponden a los calculados mediante la sintonía manual del controlador vistos anteriormente. Por otra parte, la variable options que se le pasa a lsqnonlin define los criterios para la sintonía.

La función lsqnonlin realiza 3 iteraciones como se observa en la Tabla 2, en cada una de ellas calcula los valores de la ganancia proporcional, integral y derivativa del controlador PID que minimizan la función multiobjetivo definida.

	Kp	Ki	Kd
60.8990		1.7778	0.2626
68.7955		2.4049	0.2946
69.9998		2.5	0.3

Tabla 2. Parámetros de sintonía

Con cada conjunto de valores calculados por la rutina de optimización se obtienen los errores articulares que se observan en la Figura 42.

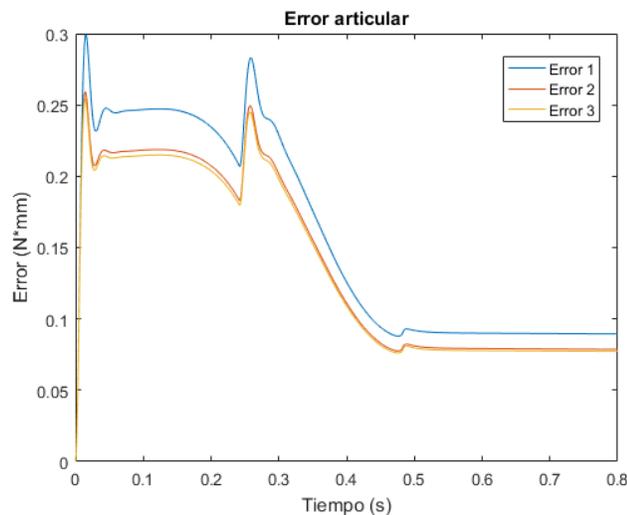


Figura 42. Error articular

Como se puede observar, a medida que el algoritmo de optimización itera y calcula un conjunto de valores para los parámetros del controlador PID, el error se

minimiza. Con los últimos valores obtenidos mediante mínimos cuadrados no lineales, el seguimiento de la trayectoria deseada se muestra en la Figura 43.

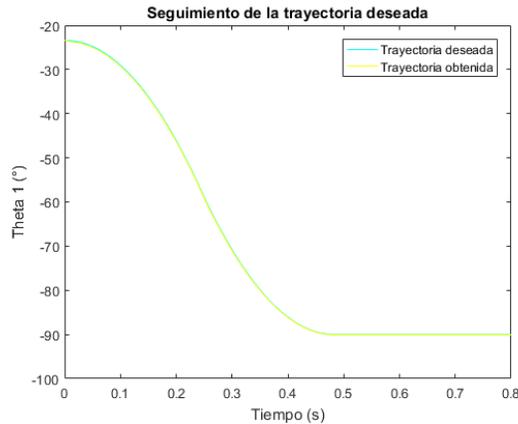


Figura 43. Seguimiento de la trayectoria articular

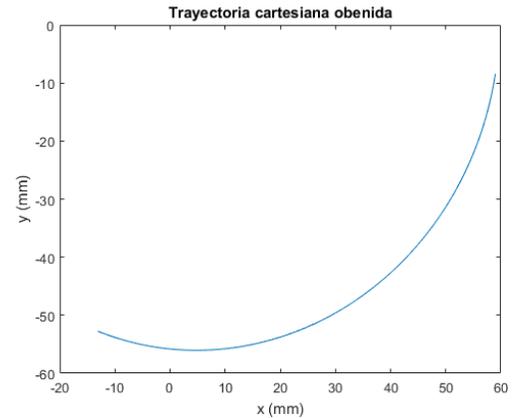


Figura 44. Trayectoria cartesiana obtenida

El error en estado estacionario con los mejores valores hallados mediante la rutina de optimización es $e_{ss} = 0,0772605516581564$ y el error final con los parámetros del controlador sintonizados manualmente es $e_{ss} = 0,123612297172997$. Por lo tanto, la sintonía del controlador PID por mínimos cuadrados no lineales es satisfactoria, permitiendo seguir con un error mínimo, la trayectoria articular deseada.

4.2 Optimización para sintonizar el control VSD

El esquema en lazo cerrado del control de posición de un dedo robótico sub-actuado se presenta en la Figura 45.

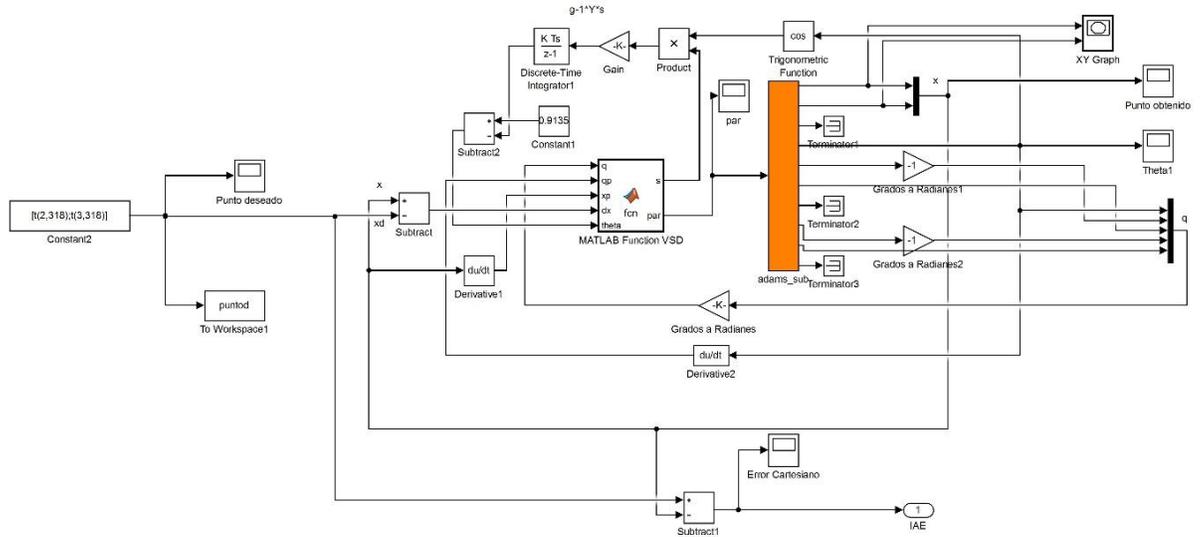


Figura 45. Esquema del control VSD

Inicialmente, los valores de los parámetros de este algoritmo de control, se sintonizan de forma manual, bajo ensayo y error, con el objetivo de tener un conjunto de valores que puedan fijarse como punto inicial, o alguno de los límites para la función de optimización, $lsqnonlin$. Los valores que se obtienen son $k = 1$, $Zita = 0.01$, $C = 0.15$, $Gamma = 0.0001$. Para este algoritmo de control natural basado en la hipótesis virtual spring damper se fija como referencia el punto deseado $x_d = [-13.1651918746829; -52.7284528744762]$. En las siguientes Figuras se observa el comportamiento del sistema en lazo cerrado

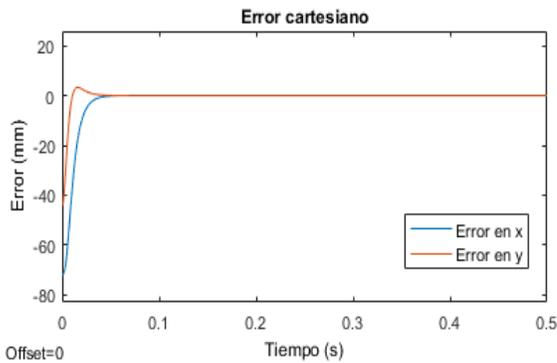


Figura 46. Error en el alcance del punto deseado

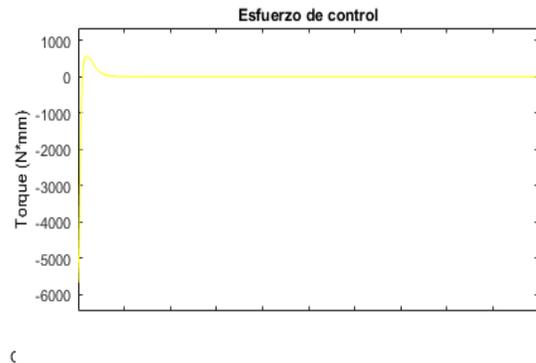


Figura 47. Esfuerzo del controlador VSD

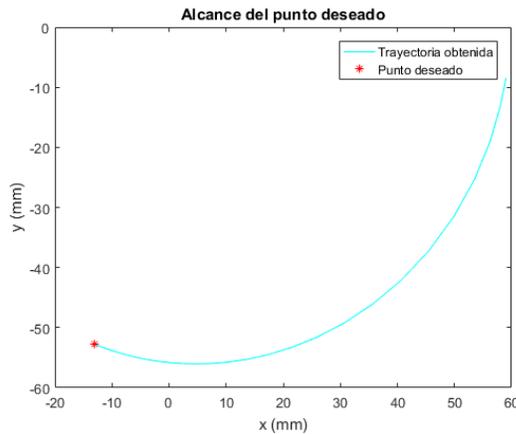


Figura 48. Trayectoria de la yema del dedo para alcanzar el punto deseado

Como se observa, con los parámetros sintonizados manualmente del controlador VSD para movimientos bajo el efecto de la gravedad, se logra alcanzar el punto deseado con un error que se aproxima a cero. Los valores de los parámetros hallados se fijan como el límite superior del algoritmo de mínimos cuadrados no lineales.

Debido a que se requiere minimizar el error entre la trayectoria cartesiana deseada y la actual para todos los pasos de tiempo de 0 a 0.5 se genera una función multiobjetivo, es decir, una función para cada paso de tiempo. El procedimiento es similar al realizado para la sintonía del controlador PID, la única diferencia son los valores iniciales de los parámetros del controlador, los cuales no corresponden a los hallados manualmente, sino a su valor medio.

La función Isqnonlin realiza 11 iteraciones como se observa en la Tabla 3, en cada una de ellas calcula los valores de los parámetros del controlador VSD que minimizan la función multiobjetivo definida.

K	Zita	C	Gamma
0.6763	0.0045	0.0602	1e-5
0.8169	0.0038	0.0465	1e-5
0.9084	0.0032	0.027	1e-5
0.9578	0.0026	0.0013	1e-5
0.9579	0.0022	0.0015	1e-5
0.9795	0.0022	7.3264e-5	1e-5
0.9848	0.0022	3.6632e-6	1e-5
0.9847	0.0042	2.3521e-6	1e-5
0.9848	0.0027	3.3354e-6	1e-5
0.9848	0.0023	3.5813e-6	1e-5
0.9847	0.0023	3.5813e-6	1.0015e-5

Tabla 3. Parámetros de sintonía VSD

Con cada conjunto de valores calculados por la rutina de optimización se obtienen los errores cartesianos que se observan en la Figura 49.

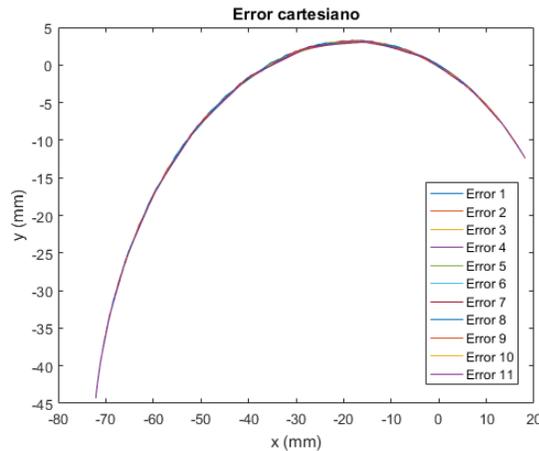


Figura 49. Errores cartesianos obtenidos

Como se puede observar, con todos los conjuntos de valores calculados por la rutina de optimización, el error es similar. Con los mejores valores obtenidos

mediante mínimos cuadrados no lineales, la trayectoria obtenida se muestra en la Figura 50.

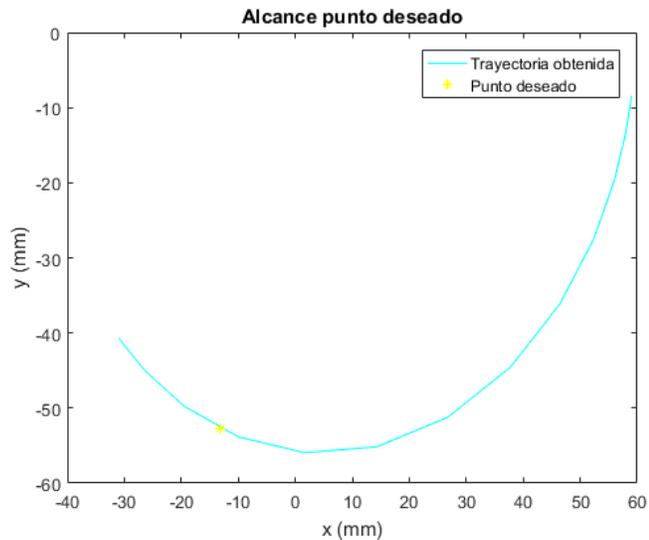


Figura 50. Alcance del punto deseado

Como se puede observar, al calcular estos valores mediante el algoritmo de optimización, la yema del dedo pasa por el punto deseado más no se detiene en él, de manera que queda un error en estado estacionario en x , $e_{ssx} = 17,521596772302$ y un error en estado estacionario en y , $e_{ssy} = -11,7365892881197$. Con los valores de los parámetros hallados manualmente se obtienen los siguientes errores en régimen permanente, $e_{ssx} = 0,0651900371876959$ y $e_{ssy} = -0,0250643422016950$. Por lo tanto, se concluye los parámetros del controlador hallados bajo ensayo y error, permiten un mejor desempeño.

5 RESULTADOS

En este capítulo se presentan los resultados que se obtuvieron con los dos controladores diseñados, considerando varias situaciones, para así poder determinar tanto sus ventajas como desventajas. Al final se analizara un último controlador, para dar cumplimiento a los objetivos trazados.

Caso 1: Análisis del desempeño del control PID articular y VSD en el seguimiento de trayectorias.

- **Desempeño control PID articular**

Para el algoritmo de control PID articular se ha definido la trayectoria de referencia que se observa en la Figura 51. Esta trayectoria tiene como punto inicial $P_0 = [-23.4^\circ]$, y finaliza en el punto $P_f = [-90^\circ]$.

Con los mejores valores obtenidos para los parámetros del controlador, los cuales fueron los hallados mediante el algoritmo de optimización, se obtiene el error de seguimiento que se observa en la Figura 52, el esfuerzo de control de la Figura 53 y el seguimiento de la trayectoria articular de la Figura 54.

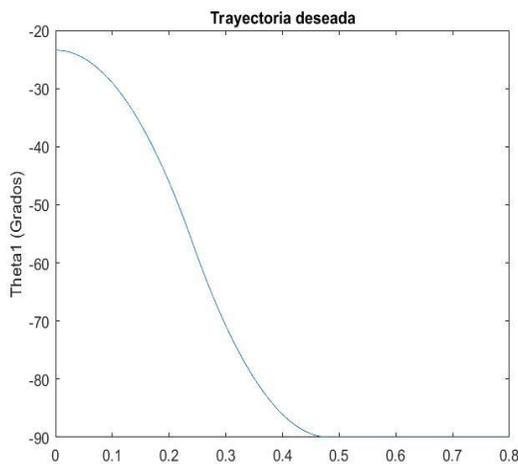


Figura 51. Trayectoria deseada

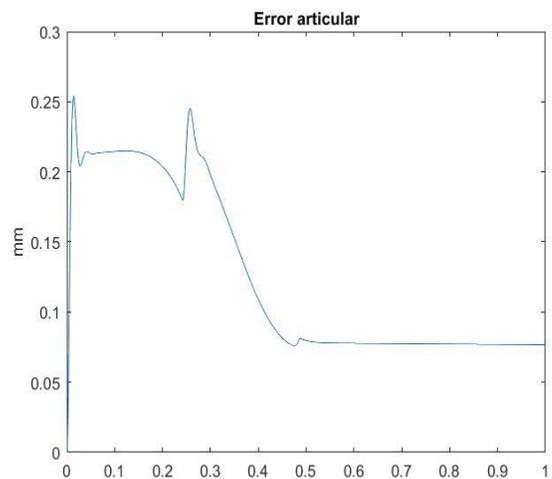


Figura 52. Error articular

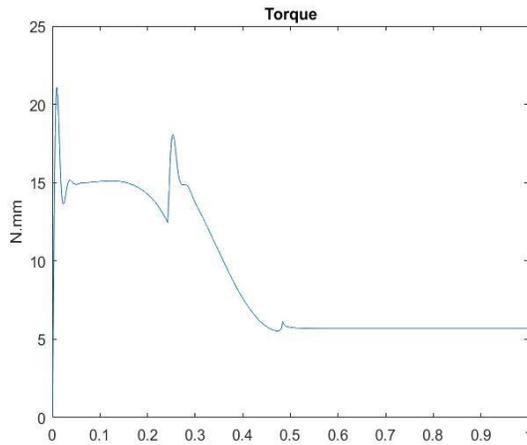


Figura 53. Esfuerzo de control

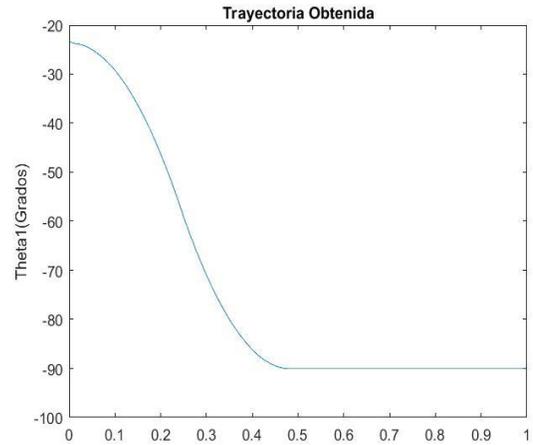


Figura 54. Seguimiento de trayectoria

Como se observa en la Figura 51, la diferencia entre la trayectoria deseada y la obtenida tiene un valor máximo de 0.2541 mm, hasta que finalmente el error en régimen permanente alcanza un valor de $e_{ss} = 0.07697$ mm. Como se ilustra en la Figura 53, el esfuerzo que realiza el controlador por seguir la trayectoria deseada, llega a un valor máximo de $u_{max} = 21.08$ N · mm. Finalmente, esta señal de control en estado estacionario es $u_{ss} = 5.681$ N · mm, este valor queda constante dado que se requiere un torque para contrarrestar los efectos de la gravedad.

- **Desempeño control VSD**

Para el algoritmo de control VSD se ha definido la trayectoria de referencia en el espacio operacional que se observa en la Figura 55. Esta trayectoria tiene como punto inicial el mismo punto que tiene la yema del dedo sin acción de control, el cual es $P_0 = [58.9999999999743; -8.42660210899956]$, y finaliza en el punto $P_f = [-13.1651918746829; -52.7284528744762]$.

Con los mejores valores obtenidos para los parámetros del controlador, los cuales fueron los hallados manualmente, se obtiene el error de seguimiento que se observa en la Figura 56, el esfuerzo de control de la Figura 57 y el seguimiento de la trayectoria cartesiana de la Figura 58.

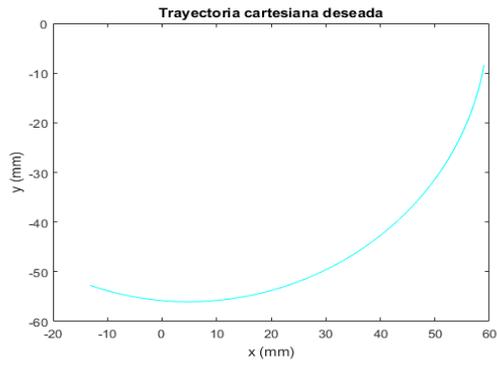


Figura 55. Trayectoria de referencia en el espacio operacional

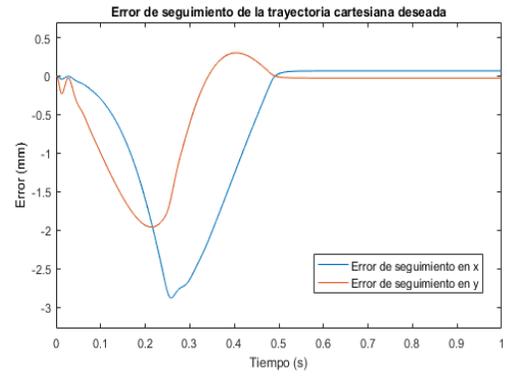


Figura 56. Error de seguimiento

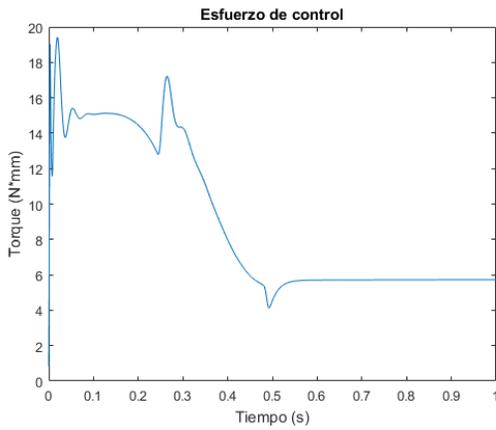


Figura 57. Torque aplicado a la articulación activa

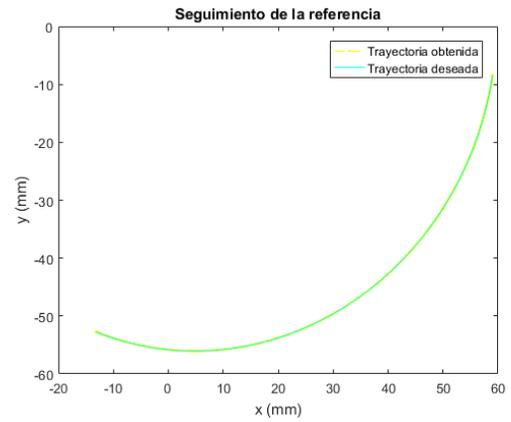


Figura 58. Seguimiento de la trayectoria deseada

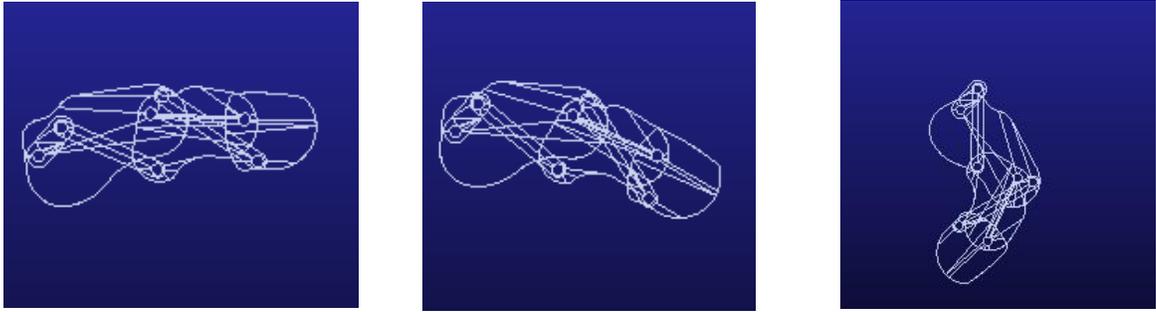


Figura 59. Simulación en Adams

Como se observa en la Figura 56, la diferencia entre la trayectoria deseada y obtenida es pequeña, hasta que finalmente el error en régimen permanente en el eje x es, $e_x = 0,0670317460783902$ mm y en el eje y , $e_y = -0,0258314400086519$ mm. Como se ilustra en la Figura 57, el esfuerzo que realiza el controlador por seguir la trayectoria deseada, llega a un valor máximo de $u_{max} = 19.2802$ N · mm. Finalmente, esta señal de control en estado estacionario es $u_{ss} = 5.7248$ N · mm, este valor queda constante dado que se requiere un torque para contrarrestar los efectos de la gravedad.

En la Figura 58 se muestra la trayectoria de la yema del dedo con respecto a la referencia definida. Como se observa, el desempeño del controlador VSD en el seguimiento de una trayectoria definida en el espacio operacional, es satisfactorio. En la Figura 59 se ilustra este seguimiento del dedo robótico sub-actuado en el entorno de simulación de Adams.

Caso 2: Análisis del desempeño del controlador PID y VSD, definiendo la referencia como un punto alcanzable por el mecanismo.

- **Desempeño control PID articular**

Se ha establecido como referencia para el controlador PID, un punto que hace parte de la trayectoria articular deseada del análisis del Caso 1, de manera que este punto es alcanzable por el mecanismo sub-actuado propuesto. Este punto deseado será $pd = -90^\circ$

Las siguientes Figuras ilustran el comportamiento del sistema en lazo cerrado.

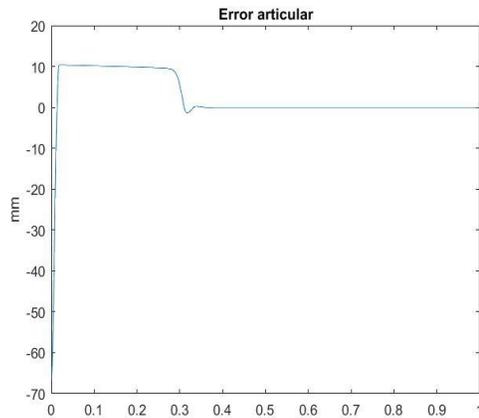


Figura 60. Error articular

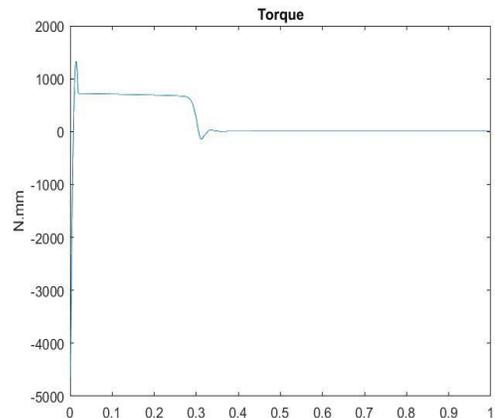


Figura 61. Esfuerzo de control

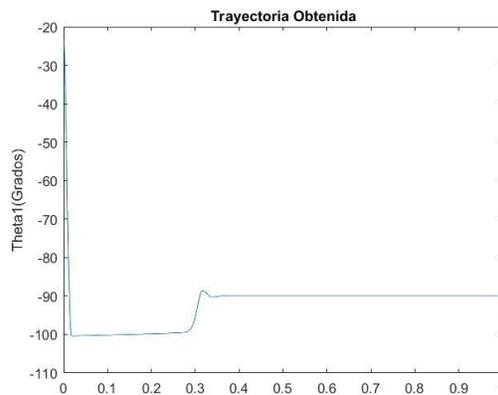


Figura 62. Trayectoria obtenida

Como se muestra en la Figura 60, el error articular alcanza un valor máximo en 10.37 mm y en estado estacionario un valor de $e_{ss} = -0.0002168$ mm.

En la Figura 61 se ilustra el esfuerzo que realiza el controlador para que la yema del dedo alcance la posición definida. Debido a que la diferencia inicial entre el punto deseado y la posición de inicio de la yema del dedo, es grande, el controlador realiza un gran esfuerzo por disminuir este error y llega a tener un

valor máximo de $u_{max} = -4661 N \cdot mm$, el torque disminuye hasta que la yema del dedo logra alcanzar la referencia y por tanto, esta señal de control toma un valor en régimen permanente de $u_{ss} = 5.731 N \cdot mm$, necesario para contrarrestar los efectos de la gravedad.

En la Figura 62 se ilustra la trayectoria que realiza la yema del dedo para alcanzar la posición deseada.

- **Desempeño control VSD**

Se ha establecido como referencia para el sistema en lazo cerrado con el control VSD, un punto que hace parte de la trayectoria cartesiana deseada del análisis del Caso 1, de manera que este punto es alcanzable por el mecanismo sub-actuado propuesto. Esta posición en el plano bidimensional a la que se quiere llegar es, $x_d = [-13.1651918746829; -52.7284528744762]$.

Las siguientes Figuras ilustran el comportamiento del sistema en lazo cerrado.

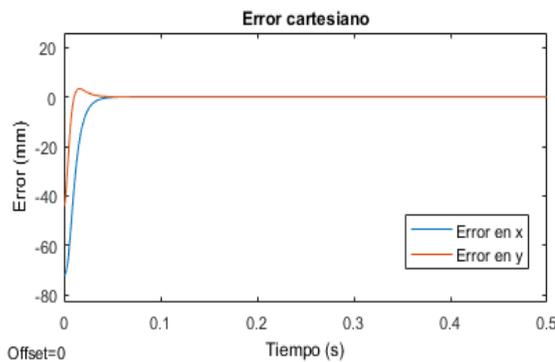


Figura 63. Error en el alcance del punto deseado

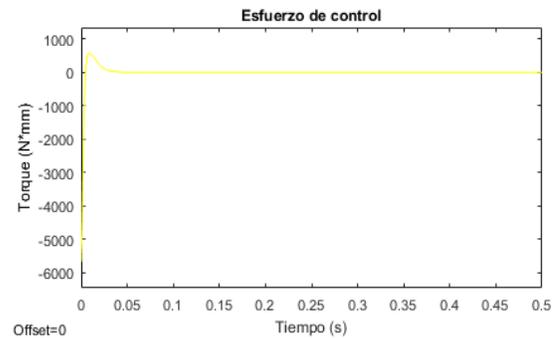


Figura 64. Esfuerzo del controlador VSD

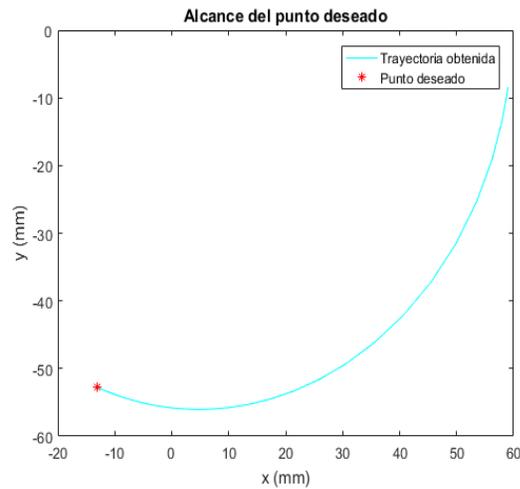


Figura 65. Trayectoria de la yema del dedo para alcanzar el punto deseado



Figura 66. Simulación en Adams.

Como se muestra en la Figura 63, el error inicial es grande debido a que el punto de inicio de la yema del dedo está alejado de la posición deseada, este error disminuye rápidamente y en estado estacionario es en el eje x es, $e_x = 0,0651900371876959$ mm y en el eje y $e_y = -0,0250643422016950$ mm.

En la Figura 64 se ilustra el esfuerzo que realiza el controlador para que la yema del dedo alcance la posición definida. Debido a que la diferencia inicial entre el punto deseado y la posición de inicio de la yema del dedo, es grande, el controlador realiza un gran esfuerzo por disminuir este error y llega a tener un valor máximo de $u_{max} = -5680,63391744183$ N · mm, el torque disminuye hasta que la yema del dedo logra alcanzar la referencia y por tanto, esta señal de control

toma un valor en régimen permanente de $u_{ss} = 5,64565426405874 \text{ N} \cdot \text{mm}$, necesario para contrarrestar los efectos de la gravedad.

En la Figura 65 se ilustra la trayectoria que realiza la yema del dedo para alcanzar la posición deseada. Como se puede observar, el desempeño del controlador VSD en el alcance de un punto deseado, es satisfactorio. En la Figura 66, se muestra el comportamiento del dedo robótico sub-actuado en el entorno de simulación de Adams.

Caso 3: Análisis del desempeño del controlador PID y VSD, definiendo la referencia como un punto inalcanzable por el mecanismo.

- **Desempeño control PID articular**

Para nuestro controlador PID un punto inalcanzable de trayectoria no estaría definido, ya que cualquier valor de ángulo para la trayectoria sería alcanzable, como ejemplo tomemos el valor de una trayectoria que va desde 0° a -360° , el dedo llegaría a su posición inicial.

- **Desempeño control VSD**

Se ha establecido como referencia para el sistema en lazo cerrado con el control VSD, un punto que no es posible de alcanzar por el mecanismo sub-actuado propuesto. Esta posición deseada es $x_d = [-40, -40]$.

En las siguientes Figuras se puede observar el comportamiento del sistema en lazo cerrado.

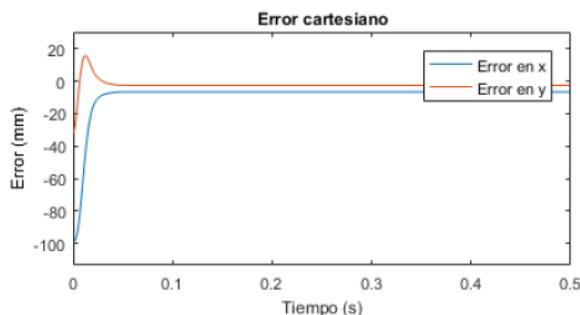


Figura 67. Error en el alcance del punto deseado

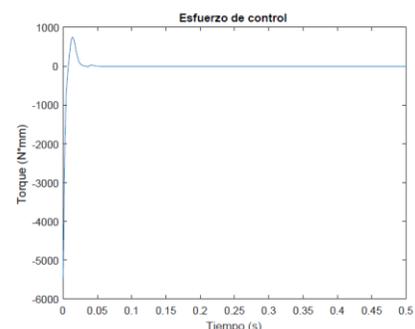


Figura 68. Esfuerzo del controlador VSD

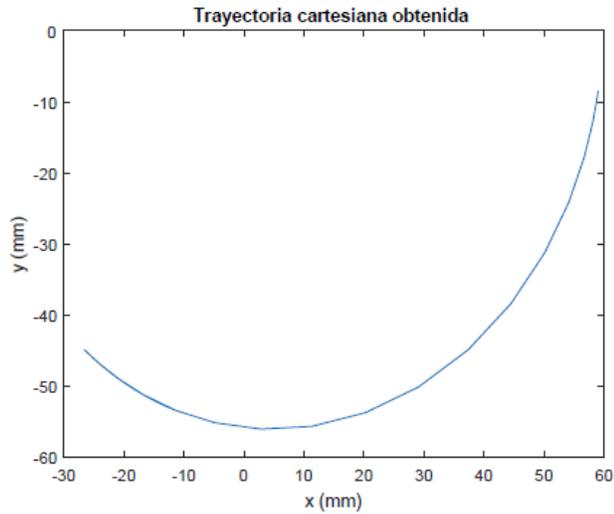


Figura 69. Trayectoria de la yema del dedo para alcanzar el punto deseado

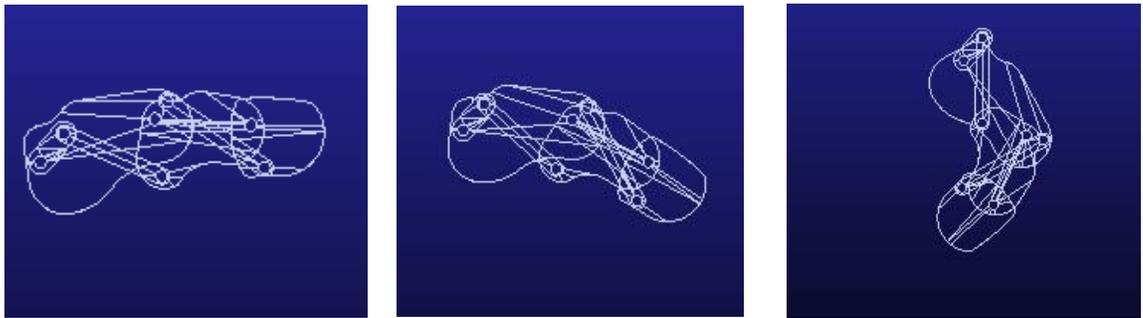


Figura 70. Simulación en Adams

Como se muestra en la Figura 67, el error inicial es grande debido a que el punto de inicio de la yema del dedo está alejado de la posición deseada, este error disminuye, sin embargo, dado que es un punto inalcanzable por el mecanismo, no desaparece.

El esfuerzo que realiza el controlador para que la yema del dedo alcance la posición definida, se observa en la Figura 68. Debido a que la diferencia inicial entre el punto deseado y la posición de inicio de la yema del dedo, es grande, el controlador realiza un gran esfuerzo por disminuir este error, tomando un valor máximo de $u_{max} = -3678,37783866716 \text{ N} \cdot \text{mm}$, lo que genera que el error disminuya. Sin embargo, dado que el punto es inalcanzable y por tanto el error no

desaparece, el controlador en lugar de incrementar el torque aplicado a la articulación activa, solo toma un valor en régimen permanente de $u_{ss} = 16,5542038424645N \cdot mm$, debido al error existente y al requerido para contrarrestar el efecto de la gravedad. El error en régimen permanente es en el eje x , $e_x = -1,10067750776898$ mm y en el eje y , $e_y = 13,4231653339448$ mm.

En la Figura 69 se ilustra la trayectoria que realiza la yema del dedo para alcanzar la posición deseada. Como se puede observar, dado que la referencia seleccionada no hace parte de los puntos posibles de alcanzar por la yema del dedo, el mecanismo sigue una trayectoria satisfactoria y se detiene en un punto cercano al deseado. En la Figura 70, se muestra el comportamiento del dedo robótico sub-actuado en el entorno de simulación de Adams.

Caso 4: Análisis del desempeño del controlador PID y VSD, en el seguimiento de una trayectoria que es interrumpida por un objeto.

- **Desempeño control PID articular**

Para esta situación, se define la misma trayectoria articular de referencia del Caso 1, la cual se puede observar en la Figura 71. A lo largo de esta trayectoria se ha simulado la presencia de un objeto. En las siguientes Figuras se ilustra el comportamiento del sistema en lazo cerrado.

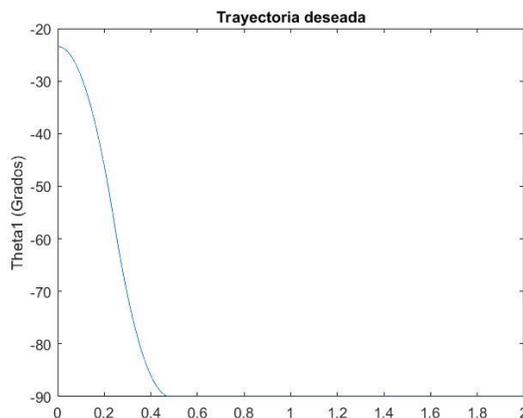


Figura 71. Trayectoria deseada

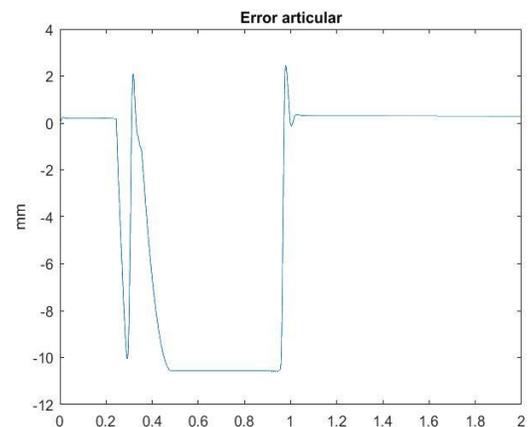


Figura 72. Error articular

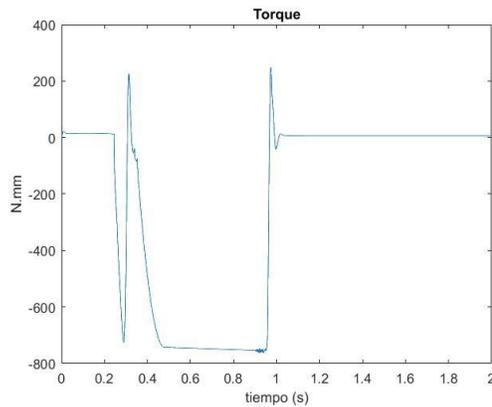


Figura 73. Esfuerzo de control

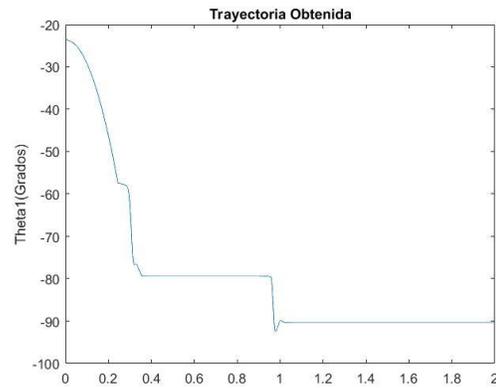


Figura 74. Trayectoria obtenida

En la Figura 72 se observa los cambios en el error articular generado por la diferencia entre la referencia y el seguimiento de esta, al encontrarse un obstáculo en el medio, alcanzando un valor en régimen permanente $e_{ss} = 0.2962 \text{ mm}$ cuando el dedo llega a su referencia deseada; en la Figura 73 se evidencia un esfuerzo de control máximo $u_{max} = -762.3 \text{ N} \cdot \text{mm}$ generado a causa del error y un esfuerzo en régimen permanente $u_{ss} = 5.539 \text{ N} \cdot \text{mm}$ cuando la trayectoria llega a su final. Por ultimo en la Figura 74 se visualiza los cambios que sufre la trayectoria que realiza el dedo, al encontrarse con el objeto, hasta que este lo saca del camino y puede completar su recorrido hasta el valor de trayectoria deseado.

- **Desempeño control VSD**

Para esta situación, se define la misma trayectoria cartesiana de referencia del Caso 1, la cual se puede observar en la Figura 75. A lo largo de esta trayectoria se ha simulado la presencia de un objeto. En las siguientes Figuras se ilustra el comportamiento del sistema en lazo cerrado.

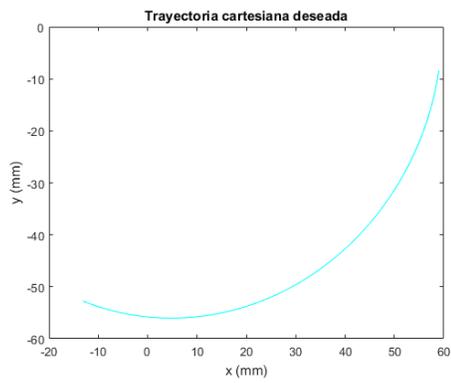


Figura 75. Trayectoria de referencia en el espacio operacional

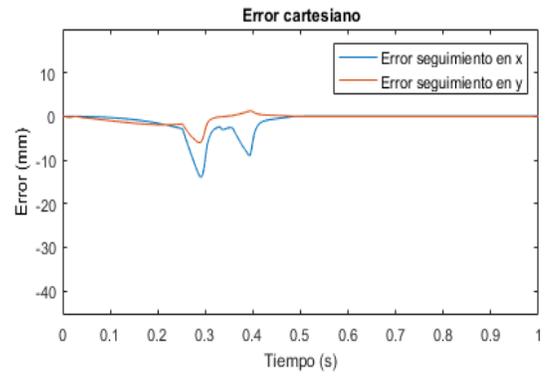


Figura 76. Error de seguimiento de la trayectoria deseada

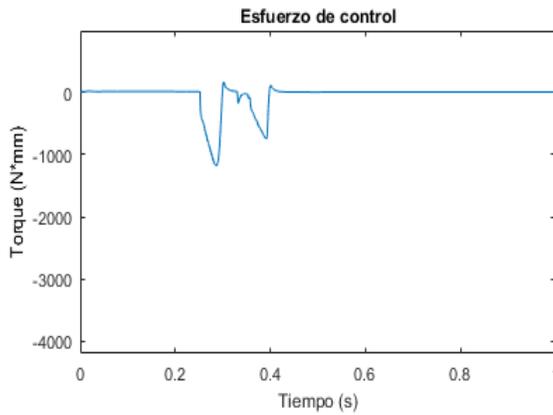


Figura 77. Torque aplicado a la articulación activa

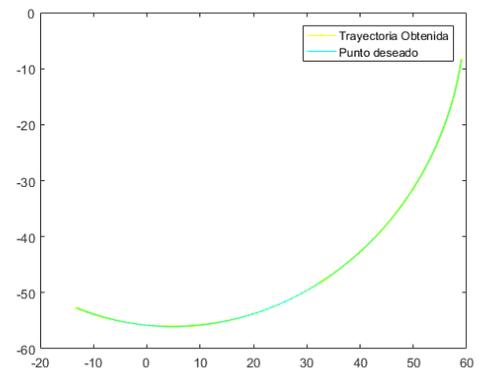


Figura 78. Seguimiento de la trayectoria deseada

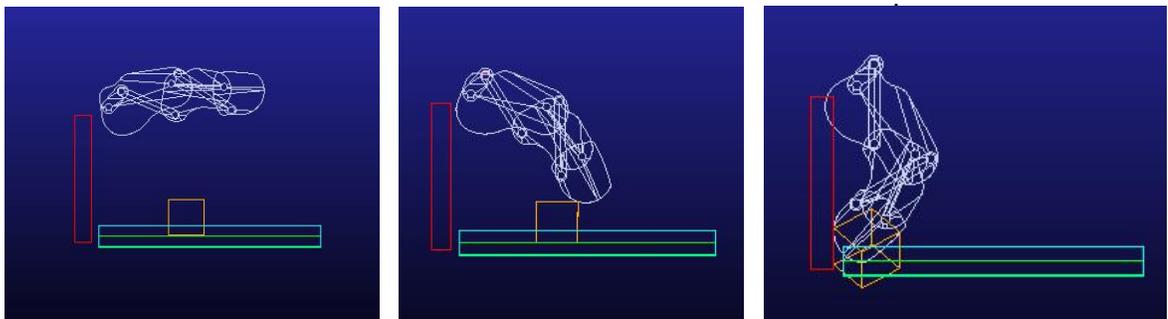


Figura 79. Simulación en Adams

Como se observa en la Figura 76, la diferencia entre la trayectoria deseada y obtenida inicialmente es pequeña, en el instante en el que la yema del dedo se tropieza con el objeto, el error aumenta, y por tanto el controlador realiza un gran esfuerzo como se ve en la Figura 77, tomando un valor máximo de $u_{max} = -1182,08794280682 \text{ N} \cdot \text{mm}$, lo que genera que el mecanismo empuje al objeto y por tanto el error disminuya, sin embargo, debido a que el torque no fue muy grande, el obstáculo no se aleja mucho de la yema del dedo, la cual vuelve a tropezar con él, y por tanto el error nuevamente aumenta, lo que implica un incremento en la señal de control, lo que genera una vez más el esfuerzo sobre el objeto, que esta vez, si se aleja lo suficiente y le permite al mecanismo retomar el seguimiento de la trayectoria deseada como se observa en la Figura 78, de esta manera el error disminuye y toma un valor en estado estacionario en el eje x , $e_x = 0,0321278735399440 \text{ mm}$ y en el eje y , $e_y = -0,0123391044231838 \text{ mm}$. Por su parte el esfuerzo de control de igual manera se reduce y toma un valor en estado estacionario de $u_{ss} = 4,57025379272832 \text{ N} \cdot \text{mm}$, requerido para contrarrestar el efecto de la gravedad. En la Figura 79 se ilustra este seguimiento del dedo robótico sub-actuado en el entorno de simulación de Adams.

Caso 5: Análisis del desempeño del controlador PID y VSD, definiendo la referencia como un punto alcanzable por el mecanismo y considerando un obstáculo.

- **Desempeño control PID articular**

Para esta situación, se establece como referencia, el mismo punto del Caso 2, el cual es alcanzable por el mecanismo sub-actuado, Además, se considera la existencia de un obstáculo entre la posición inicial de la yema del dedo y el punto deseado. En las siguientes Figuras, se ilustra el comportamiento del sistema en lazo cerrado.

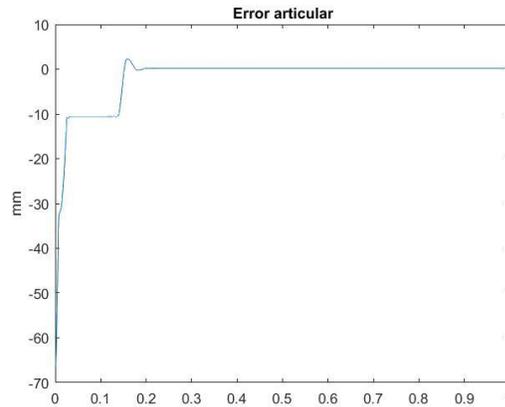


Figura 80. Error articular

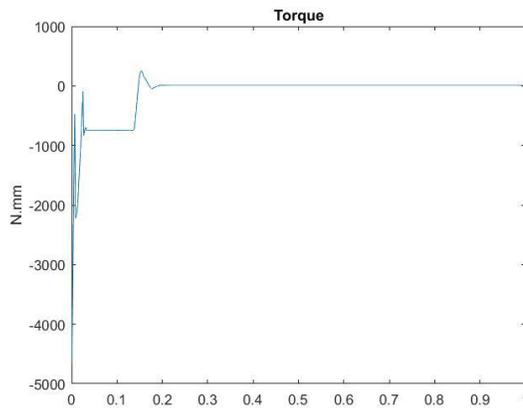


Figura 81. Esfuerzo de control

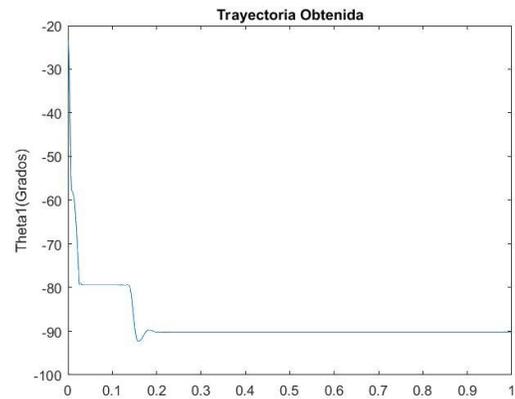


Figura 82. Trayectoria obtenida

En la Figura 80 se observa el error articular con un valor en régimen permanente $e_{SS} = 0.1474 \text{ mm}$ cuando el dedo llega a su referencia; en la Figura 81 se evidencia un esfuerzo de control máximo $u_{max} = -4661 \text{ N} \cdot \text{mm}$ a causa del error y el esfuerzo en régimen permanente $u_{SS} = 5.632 \text{ N} \cdot \text{mm}$. Por último en la Figura 82 se observa el seguimiento de la referencia en el punto alcanzable -90° y se observa en su respuesta el comportamiento que sufre al encontrarse con el obstáculo, hasta que este se escapa de su sitio y el dedo puede seguir hasta completar todo su recorrido.

- **Desempeño control VSD**

Para esta situación, se establece como referencia, el mismo punto del Caso 2, $x_d = [-13.1651918746829; -52.7284528744762]$, el cual es alcanzable por el mecanismo sub-actuado, Además, se considera la existencia de un obstáculo

entre la posición inicial de la yema del dedo y el punto deseado. En las siguientes Figuras, se ilustra el comportamiento del sistema en lazo cerrado.

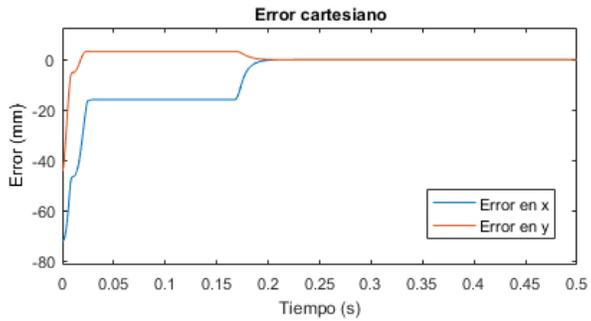


Figura 83. Error en el alcance del punto deseado

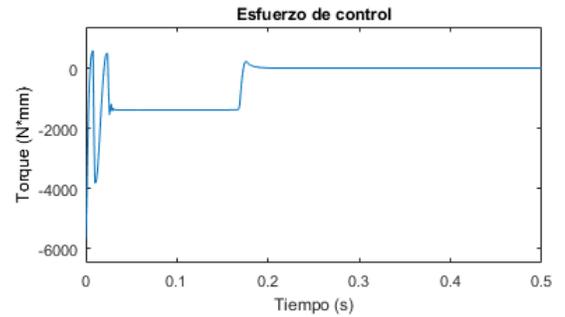


Figura 84. Esfuerzo del controlador VSD

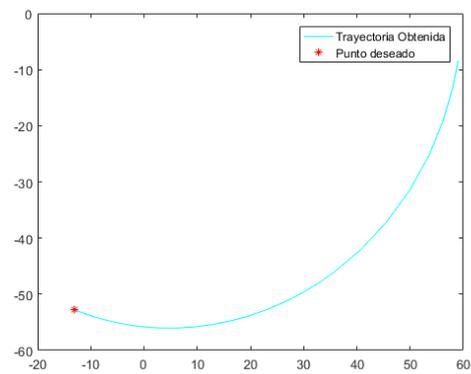


Figura 85. Trayectoria de la yema del dedo para alcanzar el punto deseado

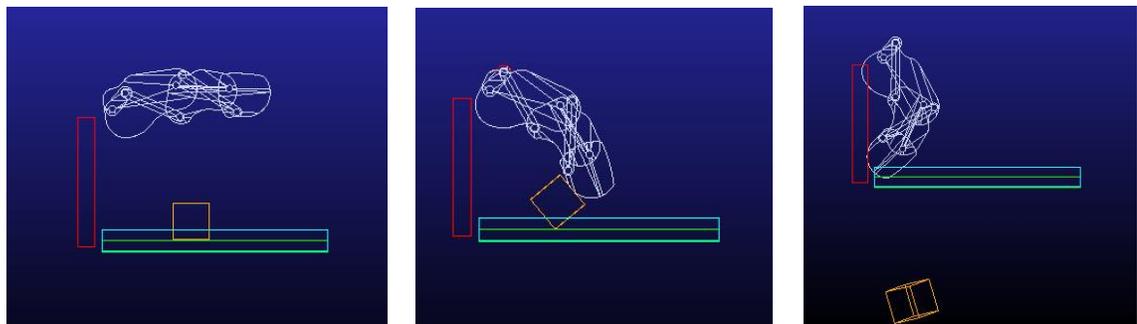


Figura 86. Simulación en Adams

Como se muestra en la Figura 83, el error inicial es grande debido a que el punto de inicio de la yema del dedo está alejado de la posición deseada, este error disminuye, por el esfuerzo del controlador visto en la Figura 84, el cual toma un valor máximo de $u_{max} = -5680,63391744184 \text{ N} \cdot \text{mm}$.

Cuando la yema del dedo se tropieza con el objeto, el error disminuye muy lentamente, por lo que nuevamente el esfuerzo de control realiza un gran esfuerzo, el error reduce su valor de manera rápida mas no desaparece por lo que la señal de control se mantiene constante en un valor grande, generando que la yema del dedo empuje el objeto, cuando lo logra, disminuye su valor y en régimen permanente toma el valor $u_{ss} = 5,64565422067018 \text{ N} \cdot \text{mm}$. El error en estado estacionario se aproxima a cero siendo, en el eje x , $e_x = 0,0651900354927726 \text{ mm}$ y en el eje y , $e_y = -0,0250643425619188 \text{ mm}$. De esta manera, la yema del dedo, efectivamente alcanza el punto deseado como se observa en la Figura 85. En la Figura 86, se muestra el comportamiento del dedo robótico sub-actuado en el entorno de simulación de Adams.

Caso 6: Análisis del desempeño del controlador PID y VSD, definiendo la referencia como un punto inalcanzable por el mecanismo y considerando un obstáculo.

- **Desempeño control PID**

Como en el caso 3 una referencia inalcanzable para el controlador PID articular, no sería posible por las razones anteriormente dichas, además, al existir un objeto en el medio, la trayectoria se vería interrumpida por este el cual impondría un límite de trayectoria alcanzable.

- **Desempeño control VSD**

Se ha establecido como referencia para el sistema en lazo cerrado con el control VSD, un punto que no es posible de alcanzar por el mecanismo sub-actuado propuesto. Esta posición deseada es $x_d = [-40, -40]$. Además, se considera un obstáculo.

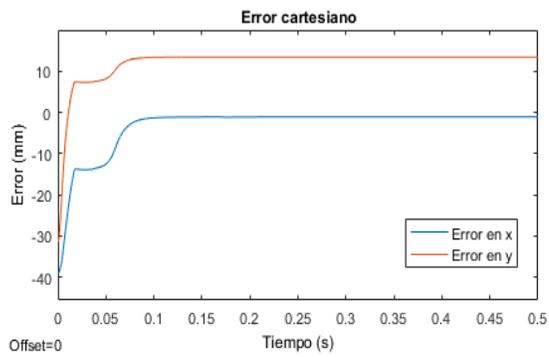


Figura 87. Error en el alcance del punto deseado

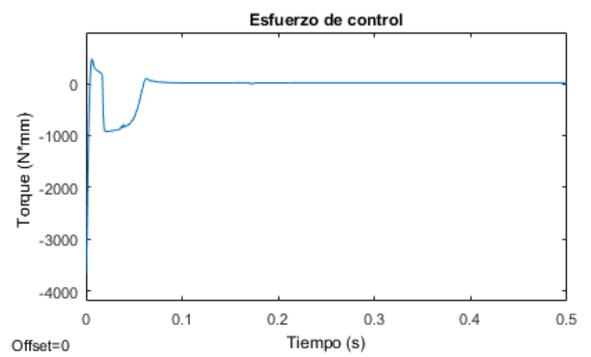


Figura 88. Esfuerzo del controlador VSD

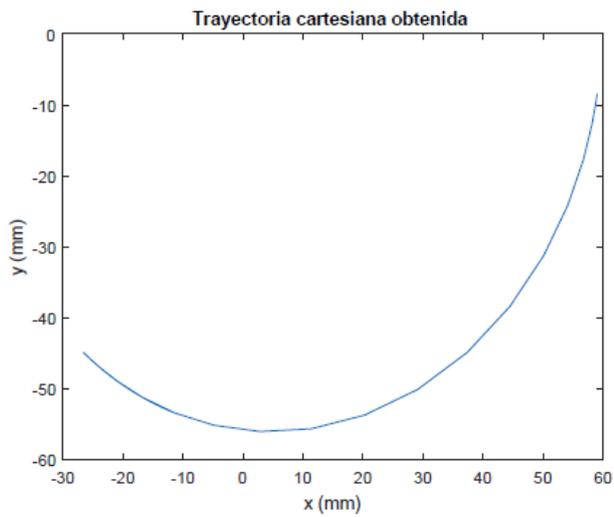


Figura 89. Trayectoria de la yema del dedo para alcanzar el punto deseado

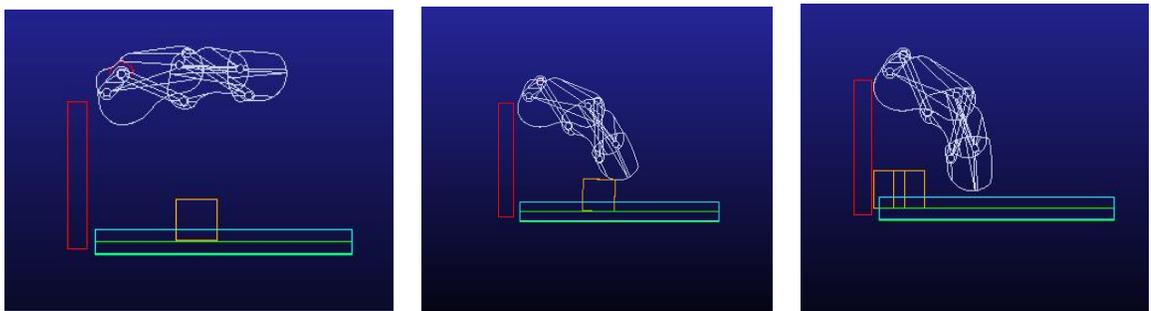


Figura 90. Simulación en Adams

Como se muestra en la Figura 87, el error inicial es grande debido a que el punto de inicio de la yema del dedo está alejado de la posición deseada, este error disminuye, por el esfuerzo del controlador visto en la Figura 88, el cual alcanza un valor máximo de $u_{max} = -3678,37783866716 \text{ N} \cdot \text{mm}$. Cuando la yema del dedo se tropieza con el objeto, el error disminuye muy lentamente, por lo que nuevamente, la señal de control toma un valor alto, lo que genera que el mecanismo empuje el objeto y por tanto el error disminuya, aunque no desaparece debido a que la posición deseada es inalcanzable, de esta manera el error toma un valor en estado estacionario en el eje x , $e_x = -1,10067762029306 \text{ mm}$ y en el eje y , $e_y = 13,4231652972234 \text{ mm}$. Por su parte el esfuerzo de control en régimen permanente es $u_{ss} = 16,5542001544225 \text{ N} \cdot \text{mm}$, el cual se debe al error existente y al requerido para contrarrestar el efecto de la gravedad.

En la Figura 89 se ilustra la trayectoria que realiza la yema del dedo para alcanzar la posición deseada. Como se puede observar, dado que la referencia seleccionada no hace parte de los puntos posibles de alcanzar por la yema del dedo, el mecanismo sigue una trayectoria satisfactoria y se detiene en un punto cercano al deseado. En la Figura 90, se muestra el comportamiento del dedo robótico sub-actuado en el entorno de simulación de Adams.

A continuación, se realiza una recopilación de los resultados obtenidos anteriormente en las siguientes tablas comparativas:

CASO 1		
	PID ARTICULAR	VSD
u_{max}	21.08 $\text{N} \cdot \text{mm}$	19.2802 $\text{N} \cdot \text{mm}$
u_{ss}	5.681 $\text{N} \cdot \text{mm}$	5.7248 $\text{N} \cdot \text{mm}$
e_{ss}	0.07697 mm	$e_{ssx} = 0.0670317460783902 \text{ mm}$ $e_{ssy} = -0.0258314400086519 \text{ mm}$

Tabla 4. Análisis del desempeño del control PID articular y VSD en el seguimiento de trayectorias.

El esfuerzo máximo que realiza el controlador PID es un poco mayor que el realizado por el VSD, con una diferencia de $1.7998 \text{ N} \cdot \text{mm}$. De igual manera el error en estado estacionario es bastante similar.

CASO 2		
	PID ARTICULAR	VSD
u_{max}	$-4661 N \cdot mm$	$-5680.63391744183 N \cdot mm$
u_{ss}	$5.731 N \cdot mm$	$5.64565426405874 N \cdot mm$
e_{ss}	$-0.000211 mm$	$e_{ssx} = 0.0651900371876959 mm$ $e_{ssy} = -0.0250643422016950 mm$

Tabla 5. Análisis del desempeño del controlador PID y VSD, definiendo la referencia como un punto alcanzable por el mecanismo.

El esfuerzo máximo que realiza el controlador PID es menor que el realizado por el VSD, con una diferencia de $1019 N \cdot mm$. De igual manera el esfuerzo de control en estado estacionario es bastante similar.

CASO 4		
	PID ARTICULAR	VSD
u_{max}	$-762.3 N \cdot mm$	$-1182.08794280682 N \cdot mm$
u_{ss}	$5.539 N \cdot mm$	$4.57025379272832 N \cdot mm$
e_{ss}	$0.2962 mm$	$e_{ssx} = 0.0321278735399440 mm$ $e_{ssy} = -0.0123391044231838 mm$

Tabla 6. Análisis del desempeño del controlador PID y VSD, definiendo la referencia como un punto inalcanzable por el mecanismo.

El esfuerzo máximo que realiza el controlador PID es menor que el realizado por el VSD, con una diferencia de $419.79 N \cdot mm$. De igual manera el error en estado estacionario es bastante similar.

CASO 5		
	PID ARTICULAR	VSD
u_{max}	$-4661 N \cdot mm$	$-5680.63391744184 N \cdot mm$
u_{ss}	$5.632 N \cdot mm$	$5.64565422067018 N \cdot mm$
e_{ss}	$0.1474 mm$	$e_{ssx} = 0.0651900354927726 mm$ $e_{ssy} = -0.0250643425619188 m$

Tabla 7. Análisis del desempeño del controlador PID y VSD, definiendo la referencia como un punto alcanzable por el mecanismo y considerando un obstáculo.

El esfuerzo máximo que realiza el controlador PID es menor que el realizado por el VSD, con una diferencia de $1019.63 N \cdot mm$. De igual manera el error en estado estacionario es bastante similar.

Controlador PID cartesiano

Como último resultado analizaremos un controlador más, este será un PID cartesiano mostrado en la Figura 91, el cual trabaja con consignas cartesianas las cuales son x e y . Se realizaron una serie de experimentos a fin de obtener resultados y conclusiones del comportamiento de este en contraste con los vistos anteriormente. Las ganancias encontradas manualmente para la sintonización de este controlador son $Kp = 0.004, Ki = 0.03, Kd = 0.04$

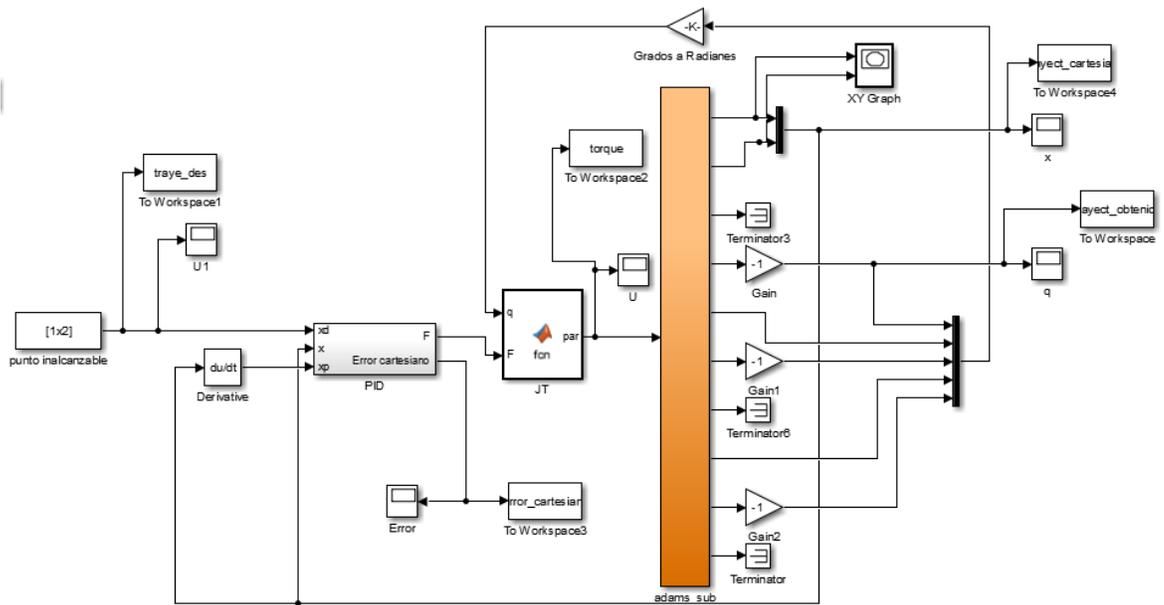


Figura 91. Diagrama de bloques controlador PID cartesiano

Caso 1: Análisis del desempeño del control PID cartesiano en el seguimiento de un punto alcanzable conocido

Para este experimento se definió un punto que fuera alcanzable, o sea, que se encuentre dentro de la trayectoria del dedo; para ello el punto escogido x e y es $(-13.1 \text{ mm}, -52.75 \text{ mm})$ en las siguientes Figuras se muestra el desempeño que se obtuvo por parte de este controlador.

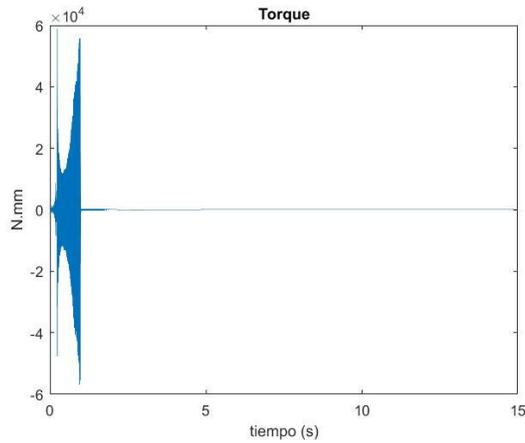


Figura 92. Esfuerzo de control

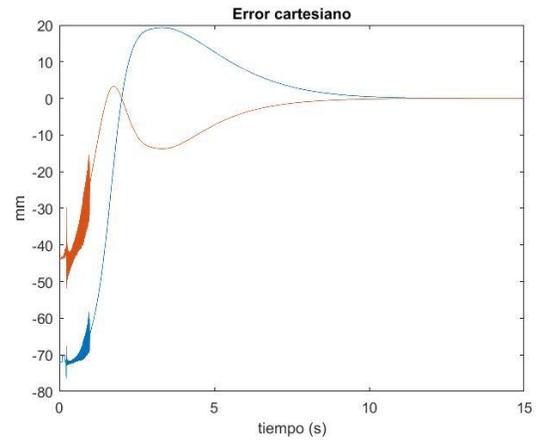


Figura 93. Error cartesiano

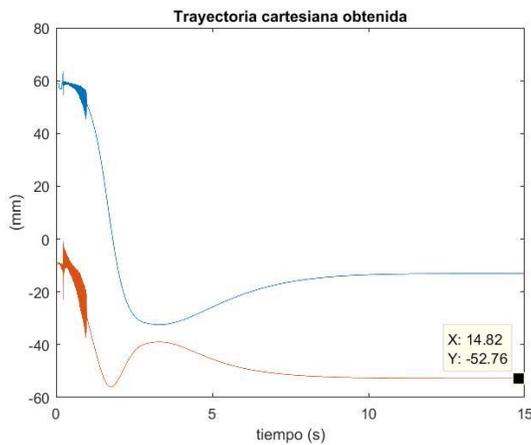


Figura 94. Seguimiento de la referencia

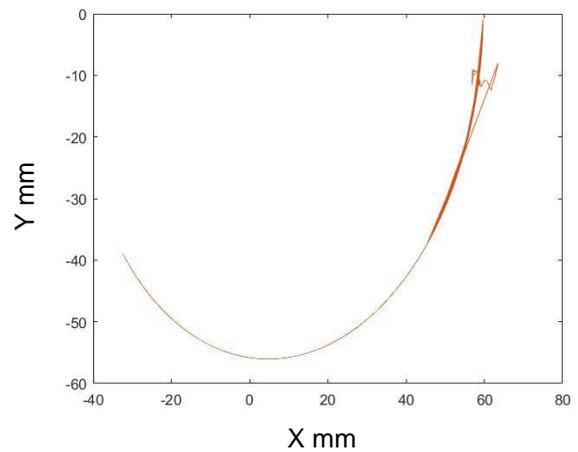


Figura 95. Trayectoria del dedo en el plano xy

Como se puede observar en la Figura 92, el esfuerzo de control inicial es bastante grande alcanzando un máximo de $-5.692 \times 10^4 N.mm$ pero a medida que logra que el error disminuya, este decrece tomando un valor en régimen estacionario de $5.681 N.mm$ una vez la referencia ha sido alcanzada. El error cartesiano presentado en la Figura 93 muestra la señal azul que representa el error en el eje x es de $-0.004532 mm$ y en el eje y representada por la señal naranja, un valor de $0.001738 mm$. En la Figura 94 se observa el seguimiento de la referencia hasta que esta alcanza su valor deseado y en la Figura 95 es mostrada la trayectoria realizada por la yema del dedo en el plano xy . También se puede observar que el

tiempo de simulación que le toma al sistema llegar a su valor deseado es un poco grande, alrededor de 15 segundos, una gran diferencia al compararlo con los dos controladores vistos anteriormente.

Caso 2: Análisis del desempeño del control PID cartesiano en el seguimiento de un punto inalcanzable

Para este segundo experimento se definió un punto que fuera inalcanzable, o sea, que se encuentre fuera de la trayectoria del dedo; para ello el punto escogido x e y es $(-40\text{mm}, -40\text{mm})$ en las siguientes Figuras se muestra el desempeño que se obtuvo por parte de este controlador.

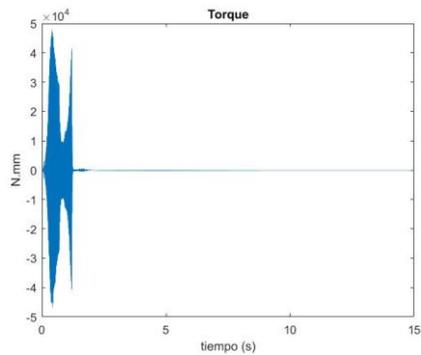


Figura 96. Esfuerzo de control

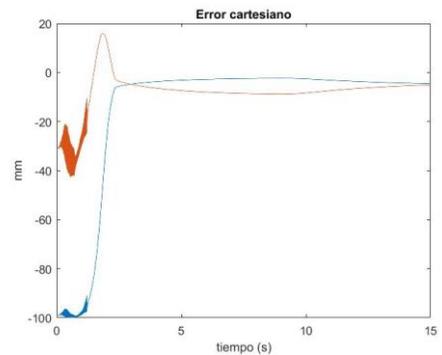


Figura 97. Error cartesiano

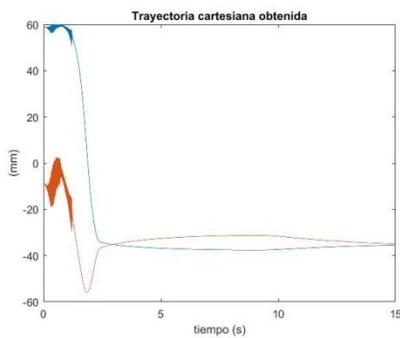


Figura 98. Seguimiento de la referencia

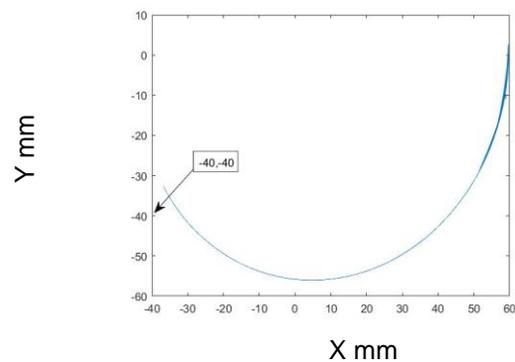


Figura 99. Trayectoria del dedo en el plano xy resaltando el punto inalcanzable escogido

Como se puede observar en la Figura 96, el esfuerzo de control inicia un poco grande alcanzando un máximo de $-4.668 \times 10^4 N.mm$ pero a medida que el error va disminuyendo este decrece, pero debido a que el punto seleccionado no es parte de la trayectoria del dedo, se observa que el controlador busca llegar a la referencia, pero mecánicamente no es posible, este se queda aplicando un torque constante de aproximadamente de $-6.939 N.mm$. En la Figura 97 se observa que el error trata de llegar a cero, pero queda en un valor constante cercano a cero, ya que al tratarse de un punto que no se encuentra en el recorrido del dedo pues este no se podrá alcanzar y por lo tanto siempre existirá error en x e y . El seguimiento de la referencia de la Figura 98 corrobora lo dicho anteriormente llegando al valor más cercano del punto escogido como inalcanzable y ahí permanece constantemente. Por ultimo en la Figura 99 se detalla el punto que esta por fuera de la trayectoria del dedo y lo que él hace al tratar de seguirlo.

Caso 3: Análisis del desempeño del control PID cartesiano en el seguimiento de un punto alcanzable considerando un obstáculo

Para este tercer experimento se definió un punto que fuera alcanzable, o sea, que se encuentre dentro de la trayectoria del dedo; para ello el punto escogido x e y es $(-13.1 mm, -52.75 mm)$, adicionalmente se agregó un obstáculo en medio de la trayectoria para observar el comportamiento del controlador a esta eventualidad. En las siguientes Figuras se muestra el desempeño que se obtuvo por parte de este controlador.

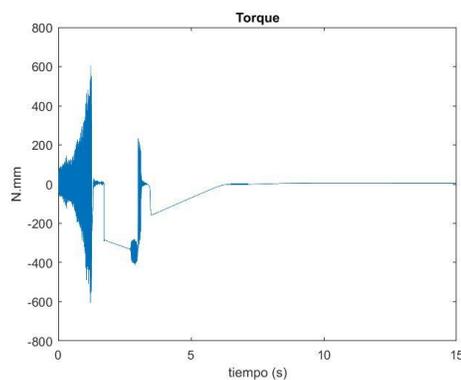


Figura 100. Esfuerzo de control

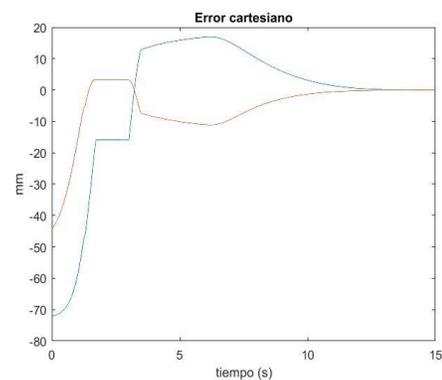


Figura 101. Error cartesiano

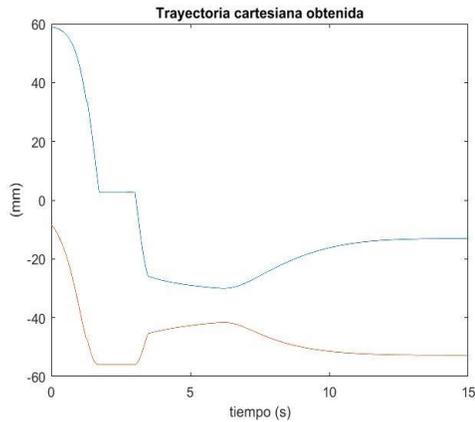


Figura 102. Seguimiento de la referencia

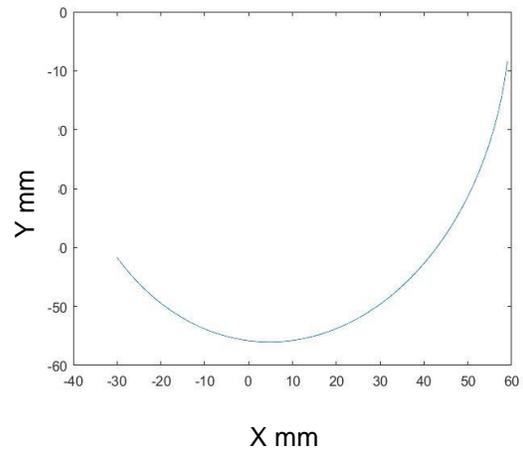


Figura 103. Trayectoria del dedo en el plano xy

Como se puede observar en la Figura 100, el esfuerzo de control inicia un poco grande alcanzando un máximo de $-603.9N.mm$ pero a medida que el error va disminuyendo este decrece tomando un valor en régimen estacionario de $5.709N.mm$ una vez la referencia ha sido alcanzada. El error cartesiano presentado en la Figura 101 muestra la señal azul que representa el error en el eje x es de $-0.004314 mm$ y en el eje y representada por la señal naranja, un valor de $0.001652 mm$. En la Figura 102 se observa el seguimiento de la referencia hasta que esta alcanza su valor deseado, no sin antes haber tenido el tropiezo con el objeto que se encuentra en una mesa construida en Adams y que simula la obstrucción en el recorrido de la trayectoria por parte del mecanismo del dedo; en la Figura 103 es mostrada la trayectoria realizada por la yema del dedo en el plano xy . También se puede observar que el tiempo de simulación que le toma al sistema llegar a su valor deseado es un poco grande, alrededor de 15 segundos, una gran diferencia al compararlo con los dos controladores vistos anteriormente.

Caso 4: Análisis del desempeño del control PID cartesiano en el seguimiento de un punto inalcanzable considerando un obstáculo

Para este cuarto experimento se definió un punto que fuera inalcanzable, o sea, que se encuentre fuera de la trayectoria del dedo, adicionalmente se agregó un obstáculo en medio de la trayectoria para observar el comportamiento del controlador a esta eventualidad.; para ello el punto escogido x e y es

$(-40\text{mm}, -40\text{ mm})$ en las siguientes Figuras se muestra el desempeño que se obtuvo por parte de este controlador.

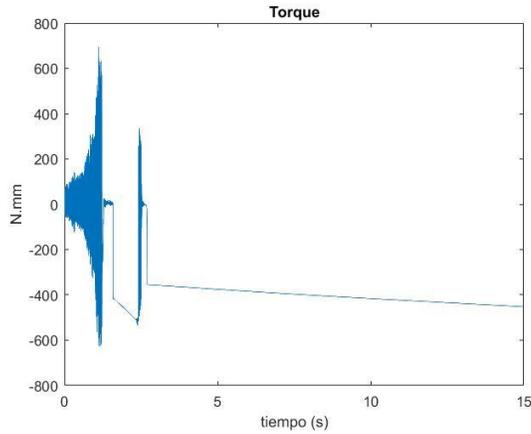


Figura 104. Esfuerzo de control

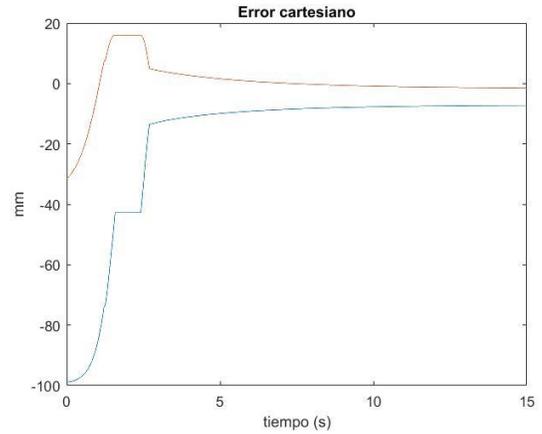


Figura 105. Error cartesiano

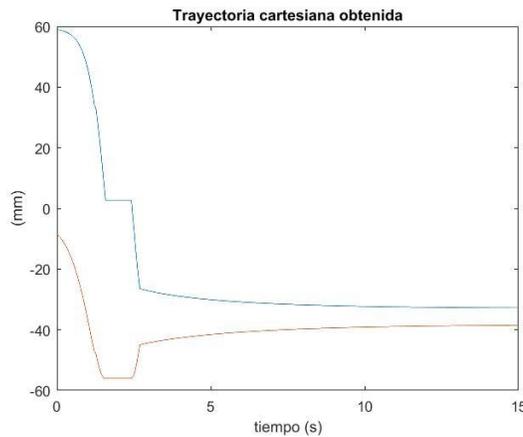


Figura 106. Seguimiento de la referencia

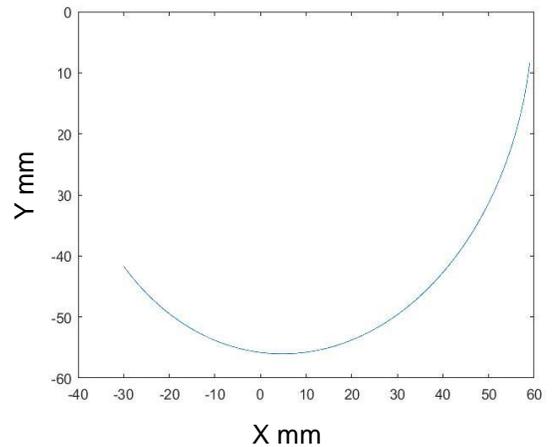


Figura 107. Trayectoria del dedo en el plano xy

Como se puede observar en la Figura 104, el esfuerzo de control inicia un poco grande alcanzando un máximo de $-626.9\text{ N}\cdot\text{mm}$ pero a medida que el error va disminuyendo este decrece; a diferencia del caso dos este es variable una vez que sobrepasa el obstáculo alcanzando un valor a los 15 segundos de simulación de $-451.2\text{ N}\cdot\text{mm}$. El error cartesiano presentado en la Figura 105 trata de llegar a cero, pero debido a que el punto de referencia es inalcanzable este se queda con un error constante a medida que el tiempo transcurre, ya que este nunca va ser cero. En la Figura 106 se observa el seguimiento de la referencia observándose cambios en esta, debido al tropiezo con el objeto que se encuentra en una mesa construida en Adams y que simula la obstrucción en el recorrido de la trayectoria

por parte del mecanismo del dedo; en la Figura 107 es mostrada la trayectoria realizada por la yema del dedo en el plano xy . También se puede observar que el tiempo de simulación es un poco grande, a diferencia de los controladores vistos anteriormente.

6 CONCLUSIONES Y TRABAJOS FUTUROS

- Los mecanismos de sub-actuación permiten generar prótesis robóticas a un bajo costo de fabricación, un consumo energético bajo y un peso más ligero. Por otra parte, la estética y el confort en los pacientes que usan este tipo de prótesis podrán ser mejoradas con tecnologías de impresión 3D, lo cual lleva a una personalización por parte del paciente.
- Una de las conclusiones a resaltar en nuestro trabajo de grado, es el hecho de que los modelos matemáticos se vuelven más complejos en cadenas cinemáticas cerradas a comparación de las cadenas cinemáticas abiertas de robots seriales, lo que lleva a un análisis más exhaustivo en la teoría de robótica existente.
- Adams es una gran herramienta computacional para realizar tipos de análisis dinámicos multi-cuerpos, ya que fue creado precisamente para trabajar problemas en este campo de aplicación. Además, observamos grandes ventajas a la hora de la aplicación a nuestro trabajo, como lo fueron la variedad de sensores para realizar mediciones, una interfaz sencilla, su plugin de control, la importación de modelos CAD prediseñados en múltiples formatos adicionalmente, la facilidad con la que se puede realizar la co-simulación con MATLAB para crear sistemas complejos de control.
- El algoritmo de control VSD presenta la ventaja de no requerir el cálculo del modelo geométrico inverso, dado que esta estrategia de control trabaja directamente en el espacio operacional y por tanto no requiere de la conversión de las coordenadas cartesianas deseadas a coordenadas articulares como si lo requieren controladores como el PID y el CTC articular. Al no implementar este modelo se minimizan los recursos computacionales requeridos y se evita el inconveniente que presenta el modelo, el cual consiste en que para alcanzar un punto en el espacio de trabajo del robot existen varias posibles soluciones para las trayectorias articulares y por tanto debe elegirse la apropiada.
- La estrategia de control VSD no se encuentra basada en el modelo matemático del robot que corresponde al modelo dinámico inverso a diferencia de controladores como el control robusto, predictivo, adaptativo, por par calculado y entre otros que si requieren el cálculo en línea de este modelo para calcular

el esfuerzo de control. Por tanto, se reduce la complejidad y los costos computacionales requeridos en la implementación de este simple y efectivo algoritmo de control.

- VSD no requiere de la definición necesaria de una trayectoria en el espacio de trabajo del sistema robótico, si se desea se puede definir solo un punto deseado.
- El algoritmo de optimización requiere la definición de un punto inicial y de límites para los valores que hacen mínima la función multiobjetivo, de manera que si no se fijan límites o el punto inicial no permite un adecuado funcionamiento del sistema, este algoritmo de mínimos cuadrados no lineales puede calcular valores para los parámetros del controlador que hacen que el sistema se vuelva inestable.
- El desempeño de un controlador PID cartesiano cuando sigue puntos conocidos que son parte de la trayectoria, es aceptable, pero tiene un inconveniente visto a lo largo de los experimentos realizados sobre este esquema de control, el tiempo que le toma al sistema alcanzar la consigna se incrementa notoriamente a diferencia del control VSD pasando de 15 segundos que le toma al PID cartesiano a 0.5 segundos, tiempo que tarda en llegar al punto deseado el controlador VSD. Dicho esto la hipótesis *Virtual Spring Damper* posee esta ventaja sobre el tradicional PID cartesiano, lo cual lo hace sobresalir muy bien.
- Cuando se trabajan sobre puntos inalcanzables por la yema del dedo, el control PID cartesiano y el VSD buscan lo mismo, aproximar el efector final a un punto dentro de la trayectoria que sea lo más cercano posible al punto en cuestión y que es imposible de alcanzar; esto se evidencia en las Figuras tomadas a lo largo de los experimentos realizados para concluir estas afirmaciones. Del esfuerzo de control se puede decir que es mal alto en el PID cartesiano teniendo torques muchísimo más altos que los ejercidos por el VSD lo cual lo coloca en desventaja frente a este.
- Al definir un punto deseado inalcanzable por la yema del dedo, el control VSD hizo que el mecanismo buscara por dos caminos opuestos alcanzar el objetivo.

- Construcción de un prototipo real de prótesis robótica de mano sub-actuada, controlada por señales electromiográficas, aplicando los conceptos desarrollados en este trabajo para el dedo índice a los demás miembros que conformarían la mano robótica.
- Validación de otro tipo de controladores y técnicas de optimización para realizar una comparación más amplia y establecer cuál sería la más óptima para este tipo de robots.
- Diseñar mecanismos sub-actuados que le permitan a la prótesis de mano robótica tener una mayor funcionalidad que la obtenida con el mecanismo de cuatro barras propuesto en este proyecto y añadir elementos pasivos que incrementen la auto adaptación del mecanismo.

BIBLIOGRAFIA

- [1] V. Kumar, "Analysis of Simple Planar Linkages," pp. 1–18, 2006.
- [2] F. C. Park and K. M. Lynch, *Introduction to robotics mechanics, planning, and control*. Cambridge University Press, 2017.
- [3] R. S. Hartenberg and J. Denavit, "A kinematic notation for lower-pair mechanisms based on metrics," *Transactions of the ASME. Journal of Applied Mechanics*, vol. 22. pp. 215–221, 1955.
- [4] C. G. Torres and L. T. Treviño, "Síntesis posicional de mecanismos doble manivela usando algoritmos evolutivos," vol. XIII, no. 48, pp. 16–21, 2010.
- [5] G. M. Freitas, A. C. Leite, and F. Lizarralde, "Kinematic control of constrained robotic systems," vol. 22, no. 6, 2011.
- [6] N. Orlandea, "Node Analogous, Sparsity Oriented Methods for Simulation of Mechanical Dynamic Systems," University of Michigan, 1973.
- [7] J. B. Mcconville, *Introduction to Mechanical System Simulation Using Adams*. Ann Arbor, Michigan: SDC Publications, 2015.
- [8] H. Zhongling, X. Dongsheng, L. Guangsheng, and J. Changzhi, "Study on Co-Simulation Technology Based on ADAMS and MATLAB," no. Ism3e, pp. 595–598, 2015.
- [9] F. Cheraghpour and P. Branch, "Dynamic Modeling and Kinematic Simulation of Staubli© TX40 Robot Using MATLAB/ADAMS Co-simulation," in *International Conference on Mechatronics*, 2011, pp. 386–391.
- [10] Wang, J., Gosselin, C. and Cheng, L. (2002). *Multibody System Dynamics*, 7(2), pp.145-170.
- [11] O. Vivas Albán, *Diseño y control de robots industriales*. Buenos Aires: Elaleph.com, 2010.
- [12] S. Arimoto and M. Sekimoto, "Human-like movements of robotic arms with redundant DOFs: virtual spring-damper hypothesis to tackle the Bernstein problem," *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, Orlando, FL, 2006, pp. 1860-1866.
- [13] S. Arimoto, J.-H. Bae, H. Hashiguchi, and R. Ozawa, "Natural resolution of ill-posedness of inverse kinematics for redundant robots under constraints," *Communications in Information and Systems*, vol.4, no.1, pp.1–28, 2004.

- [14] S. Arimoto, M. Sekimoto, H. Hashiguchi, and R. Ozawa, "Natural resolution of ill-posedness of inverse kinematics for redundant robots: A challenge to Bernstein's degrees-of-freedom problem," *Advanced Robotics*, vol.19, no.4, pp.401–434, 2005.
- [15] S. Arimoto, H. Hashiguchi, and R. Ozawa, "A simple control method coping with a kinematically ill-posed inverse problem of redundant robots: Analysis in case of a handwriting robot," *Asian Journal of Control*, Vol. 7, No. 2, pp. 112–123, 2005.
- [16] A. Fassih, D. Subbaram Naidu, S. Chiu and M. P. Schoen, "Precision grasping of a prosthetic hand based on virtual spring damper hypothesis," *2010 5th Cairo International Biomedical Engineering Conference*, Cairo, 2010, pp. 79-82.
- [17] J. T. Kim, J. S. Cho, S. H. Whang, S. U. Chon and S. Park, "Method to apply virtual spring-damper hypothesis for a redundant hydraulic manipulator," *2014 11th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Kuala Lumpur, 2014, pp. 454-457.
- [18] J. h. Bae, M. Sekimoto and S. Arimoto, "Effect of Virtual Spring-Damper in Grasping and Object Manipulation of a Robotic Hand-Arm System," *2006 SICE-ICASE International Joint Conference*, Busan, 2006, pp. 2222-2226.
- [19] S. K. Kim, J. H. Bae, Y. Oh and Sang-Rok Oh, "Concurrent control of position/orientation of a redundant manipulator based on virtual spring-damper hypothesis," *2011 IEEE International Conference on Robotics and Automation*, Shanghai, 2011, pp. 6045-6050.
- [20] Fassih, A., Naidu, D., Chiu, S. and Schoen, M. (2010). Power Grasping of a Prosthetic Hand based upon Virtual Spring-Damper Hypothesis. *IASTED Technology Conferences / 705: ARP / 706: RA / 707: NANA / 728: CompBIO*.
- [21] Es.mathworks.com. (2017). *Solve nonlinear least-squares (nonlinear data fitting) problems - MATLAB lsqnonlin - MathWorks United Kingdom*. [online] Available at: <https://es.mathworks.com/help/optim/ug/lsgnonlin.html>.