



# TESIS

Universidad La Salle Noroeste

“Desarrollo de un modelo representativo de redes neuronales y experimentación de un modelo mejorado, para la solución del problema cinemático inverso de un mecanismo 5R”

*Que para obtener el título de*  
**Licenciatura en Ingeniería Mecatrónica**

*con Reconocimiento de Validez Oficial No. 2012247  
de fecha 17 de agosto de 2001*

*presenta*

**Renée Nicole Espinoza Miranda**

*Asesor de Contenido  
Dr. Eusebio Jiménez López*

Ciudad Obregón, Sonora, México; febrero 2020

ULSA – Biblioteca  
Tesis Digitales  
Restricciones de uso

DERECHOS RESERVADOS ©

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

El uso para la reproducción parcial o total del contenido que sea objeto de protección de derechos de autor, se restringe para fines educativos e informativos, citando la fuente de donde se obtuvo y sus autores.

Sanciones pueden resultar aplicables en caso de lucro, reproducción, edición o modificación, sin la autorización del titular de los derechos.

## **DEDICATORIAS**

La presente investigación la dedico a mis padres Maribel Miranda y Víctor Hugo Espinoza, al igual que a mi hermana Olga Alexia Espinoza, quienes son mi ejemplo a seguir además de que me dieron su apoyo incondicional, así como sus enseñanzas.

A Dios que estuvo en cada momento a lo largo de la investigación además de permitirme concluir el presente trabajo, así como guiarme por el mejor camino para mi persona.

Asimismo, dedico este trabajo de investigación a todas las personas que me apoyaron y guiaron a lo largo del trabajo, además de que espero sea útil para las futuras generaciones.

## **AGRADECIMIENTOS**

Agradezco principalmente a mi familia quienes me brindan su apoyo incondicional en cada paso que doy.

Agradezco especialmente a mi asesor Dr. Eusebio Jiménez López, así como mi Co-asesor Kevin Fierro Ruiz por el tiempo dedicado a la presente investigación, así como sus consejos y entusiasmo.

También, quiero agradecer a los ingenieros Daniel Servín De La Mora y Raúl Servín De La Mora quienes hicieron grandes aportaciones a esta investigación.

Asimismo, agradezco a mis profesores por brindarnos las bases y el conocimiento necesario para enfrentar los retos venideros en la industria.

Por último, quiero agradecer a Universidad La Salle Noroeste por todo su apoyo.

## RESUMEN

La Inteligencia Artificial es un campo de las ciencias computacionales que ha tomado relevancia hoy en día en las aplicaciones de ingeniería, particularmente en los campos de la mecánica y la robótica. Las redes neuronales son utilizadas para un sin número de aplicaciones en la robótica, entre ellas en el estudio de la cinemática, el control y el diseño. En este trabajo de tesis se analiza el problema cinemático inverso relacionado con un robot delta RR o también conocido como mecanismo paralelo 5R, con el propósito de entrenar una red neuronal representativa que sea capaz de calcular los ángulos y los ejes de rotación de los eslabones que componen el robot dada la información de entrada de una trayectoria o sucesión de puntos. Posteriormente, con el modelo representativo y considerando 10000 datos de entrada y salida, se realizó una experimentación computacional para determinar una topología “especial de red neuronal” considerando un total de 81 experimentos generados por 4 variables con 3 niveles de cambios. Las variables fueron: Número de Capas: (2, 3, 4), Número de Neuronas por capa (3, 4, 5), Número de Épocas (1000, 2000, 3000) y Rendimiento:( 0.001, 0.0001, 0.00001). Se identificó que la mejor topología fue la de 2 capas, con 3 neuronas, 2000 épocas y un rendimiento de .00001, ya que el ajuste logrado fue lo suficiente bueno para aproximar el sistema de la cinemática inversa con un  $\pm 1.5\%$  de tolerancia en el cálculo del modelado matemático. Por lo que se puede concluir que fue posible entrenar una red neuronal especial para el cálculo de la cinemática inversa del mecanismo considerado.

## ÍNDICE

<b>CAPÍTULO I: INTRODUCCIÓN</b> .....	11
1.1 Antecedentes .....	11
1.2 Planteamiento del problema .....	18
1.3 Pregunta de investigación.....	20
1.4 Objetivos .....	20
1.4.1 Objetivo general.....	20
1.4.2 Objetivos específicos.....	20
1.5 Justificación .....	21
1.6 Delimitaciones.....	21
<b>CAPÍTULO II: MARCO TEÓRICO</b> .....	22
2.1 Robótica .....	23
2.1.1 Definición de robot .....	23
2.1.2 Clasificación de robots .....	24
2.1.3 Aplicaciones de los robots.....	24
2.1.4 Robots paralelos.....	28
2.2 Cinemática de los robots.....	34
2.2.1 Cadenas cinemáticas.....	35
2.2.2 Hipótesis de cuerpo rígido .....	35
2.2.3 Trayectoria .....	36
2.2.4 Interpolación .....	37
Interpolar hace referencia a encontrar un valor el cual se encuentre entre dos o más puntos base conocidos, los cuales se aproximan por medio de polinomios. Existen dos tipos de interpolación la que es con espacios equidistantes y con espacios no equidistantes. ....	37
2.2.5 Formulación de los problemas directo e inverso.....	41
2.3 Álgebra de números complejos .....	48
2.3.1 Parejas ordenadas de números reales y propiedades de los números complejos.....	50
2.3.2 Caracterización de las rotaciones con números complejos.....	53
2.4 El método de Newton-Raphson.....	56
2.4.1 Análisis para dos variables .....	59

2.5 Inteligencia artificial .....	62
2.5.1 Métodos de inteligencia artificial.....	63
2.5.2 Aplicaciones de la inteligencia artificial .....	69
2.5.3 Las redes neuronales.....	72
2.6 Lenguajes de Programación .....	77
2.6.1 Python .....	79
2.6.2 Lenguaje M.....	80
<b>CAPÍTULO III: METODOLOGÍA .....</b>	<b>81</b>
3.1 Tipo de investigación. ....	81
3.2 Diseño de la investigación.....	82
3.3 Variables de estudio.....	83
3.4 Objetos de estudio. ....	83
3.5 Materiales e instrumentos.....	84
3.5.1 MATERIALES .....	84
3.5.2 INSTRUMENTOS .....	84
3.6 Procedimiento.....	84
<b>CAPÍTULO IV: RESULTADOS Y DISCUSIÓN. ....</b>	<b>90</b>
4.1 Modelado del mecanismo.....	90
4.1.1 Modelado del mecanismo: configuración de referencia o inicial .....	90
4.1.2 Modelado de la configuración final. ....	94
4.1.3 Modelado de desplazamiento. ....	98
4.2 Entrenamiento de una red neuronal artificial. ....	100
4.2.1 Aplicación de los protocolos. ....	102
4.2.1.1 Protocolo para obtener los datos de entrada para la red neuronal .....	102
4.2.2 Experimentación para seleccionar una topología especial.....	125
<b>CAPÍTULO V: CONCLUSIONES.....</b>	<b>149</b>
<b>REFERENCIAS .....</b>	<b>155</b>

## ÍNDICE DE FIGURAS

Figura 1. Primera aplicación industrial de los Robots Paralelos patentada en 1942. (Díaz, 2018)...	14
Figura 2. Plataforma Stewart usada por Lufthansa como simulador de vuelo (Díaz, 2018).....	15
Figura 3. Representación esquemática del robot Delta (Díaz, 2018). .....	16
Figura 4. Esquema de funcionamiento de un robot híbrido donde el robot paralelo provee de 3 ejes (Díaz, 2018). .....	17
Figura 5. Robot industrial (Kumar Saha, 2011) .....	26
Figura 6. Mars Rovers. (National Geographic, 2021) .....	27
Figura 8. Plataforma de Gough (M. Díaz-Rodríguez, 2018) .....	32
Figura 9. Robot paralelo P-arm utilizado en cirugía laparoscópica (M. Díaz-Rodríguez, 2018).....	33
Figura 10. Sistema de Coordenadas Dextrógiro (s. a.,2016).....	42
Figura 11. Representación simbólica y asignación de marcos de referencia (Cardoso, 2017). .....	43
Figura 12. Representación geométrica. (Cardoso, 2017).....	44
Figura 13. Esquemático de los 3 primeros grados de libertad del robot. (Rodríguez., s.f.).....	45
Figura 14. Configuraciones codo arriba y abajo. (Rodríguez, s.f) .....	47
Figura 15. Definición de catetos e hipotenusa. (Ortiz, 2008). .....	49
Figura 16. Rotación del ángulo $\theta$ (Riviera, 2001). .....	54
Figura 17. Rotación del plano. (Riviera F., 2001). .....	56
Figura 18. Método de Newton Raphson. (Chaillou, 2008).....	58
Figura 19. Proceso de evolución de los algoritmos genéticos. (Campos M. 2019).....	65
Figura 20. Red Neuronal Artificial. (Costa A., 2021).....	66
Figura 21. Sistema basado en lógica difusa. (Fronzino E. 2017) .....	67
Figura 22. Ejemplo de una red neuronal totalmente conectada. (Matich, D. J. (2001, marzo). <i>Redes Neuronales: Conceptos Básicos y Aplicaciones</i> . Universidad Tecnológica Nacional – Facultad Regional Rosario.) .....	74
Figura 24. Diagrama de desarrollo. (Espinoza, 2021) .....	85
Figura 25. Configuración del mecanismo paralelo plano 5R y bases de la configuración inicial o no deformada (Jiménez <i>et al.</i> 2021b). .....	91
Figura 26. Bases móviles en la configuración final. (Jiménez et al, 2021b) .....	94
Figura 27. Desplazamientos angulares en la configuración inicial y final (Jiménez et al, 2021b).....	97
Figura 28. Localización de la zona de recolección de las coordenadas de los puntos en el área de trabajo.....	103

Figura 29. Zona de localización de los puntos.....	104
Figura 30. Malla de puntos. ....	105
Figura 31 Configuraciones espejo del robot .....	109
Figura 32. Configuraciones para alcanzar el punto (89.9, 76.6) .....	109
Figura 33. Topología de la red neuronal para el modelo Representativo. ....	110
Figura 34. Topología y algoritmo utilizado para entrenamiento de la red. ....	112
La figura 35 muestra la pantalla de Matlab con los resultados de la corrida de datos. ....	113
Figura 35. Pantalla de Matlab con los resultados de la corrida de datos. ....	113
Figura 36. Rendimiento red neuronal. ....	114
Figura 37. Estado de entrenamiento de la red neuronal. ....	115
Figura 38. Error de Histograma de la red neuronal. ....	116
Figura 39. Estado de Regresión de la red neuronal. ....	116
Figura 40. Error entre Red Neuronal y Newton Rapshon (ph1).....	119
Figura 41. Error entre Red Neuronal y Newton Rapshon (ph2).....	120
Figura 42. Error entre Red Neuronal y Newton Rapshon (ph3).....	120
Figura 43. Error entre Red Neuronal y Newton Rapshon (ph4).....	121
Figura 44. Error entre Red Neuronal y Newton Rapshon (P0).....	121
Figura 45. Error entre Red Neuronal y Newton Rapshon (P1).....	122
Figura 46. Error entre Red Neuronal y Newton Rapshon (Q0). ....	122
Figura 47. Error entre Red Neuronal y Newton Rapshon (Q1). ....	123
Figura 48. Error entre Red Neuronal y Newton Rapshon (R0).....	123
Figura 49. Error entre Red Neuronal y Newton Rapshon (R1).....	124
Figura 50. Error entre Red Neuronal y Newton Rapshon (S0). ....	124
Figura 51. Error entre Red Neuronal y Newton Rapshon (S1). ....	125
Figura 52. Pantalla frontal donde se muestra el entrenamiento y las principales variables para cada experimento.....	131
La Tabla 13 muestra el primer filtrado el cual está relacionado con el gradiente.....	134
Figura 53. Entrenamiento para la red [2, 4, 2000, 0.00001].....	137
Figura 54. Gráfica del error y el número de épocas de la red [2, 4, 2000, 0.00001]. ....	137
Figura 55. Parámetros de la red entrenada .....	138
Figura 56. Histograma de errores .....	139
Figura 57. Ajuste de datos y ganancia.....	140

Figura 58. Entrenamiento para la red [3, 4, 2000, 0.00001].	142
Figura 59. Gráfica del error y el número de épocas de la red [3, 4, 2000, 0.00001].	142
Figura 60. Parámetros de la red entrenada [3, 4, 2000, 0.00001].	143
Figura 61. Histograma de errores de la red entrenada [3, 4, 2000, 0.00001].	143
Figura 62. Ajuste de datos y ganancia de la red entrenada [3, 4, 2000, 0.00001].	144
Figura 63. Best training Performance: 1) Red [2, 4, 2000, 0.00001] y 2) Red [3, 4, 2000, 0.00001]	145
Figura 64. Training State: 1) Red [2, 4, 2000, 0.00001] y 2) Red [3, 4, 2000, 0.00001]	146
Figura 65. Error histogram: 1) Red [2, 4, 2000, 0.00001] y 2) Red [3, 4, 2000, 0.00001]	147
Figura 66. Regression: 1) Red [2, 4, 2000, 0.00001] y 2) Red [3, 4, 2000, 0.00001]	148

## ÍNDICE DE TABLAS

Tabla 1. Variable de estudio (Espinoza, 2021) .....	83
Tabla 2 Tabla de coordenadas de los puntos sobre la zona de entrenamiento del robot .....	106
Tabla 4. Cálculo para 10 puntos de 10000 puntos de la cinemática inversa. ....	107
Tabla 5. Cálculo para 20 puntos de 10000 puntos de la cinemática inversa. ....	108
Tabla 6. Tabla que representa las combinaciones de topologías que se usaron en el modelo representativo, para 20 puntos. ....	111
Tabla 7. Tabla de parámetros durante el entrenamiento del modelo representativo, para 20 datos. ....	114
Tabla 8. Tabla de validación .....	117
Tabla 9. Entradas y salidas con método Newton Rapshon, muestra de 20 puntos.....	118
Tabla 10. Entradas y salidas con red neuronal, muestra de 20 puntos. ....	118
Tabla 11. Error entre red Neuronal y método de Newton Rapshon.....	119
Tabla 12. Información de los 81 experimentos realizados en Matlab.....	133
Tabla 13. Primer filtrado en base al gradiente.....	134
Tabla 14. Primer filtrado en base al rendimiento .....	134
Tabla 15. Tercer filtrado en base al target.....	135
Tabla 16. Tercer filtrado en base al target.....	135
Tabla 17. Cuarto filtrado en base al target .....	136
Tabla 18. Comparativa entre el N-R y AI. ....	141
Tabla 19. Comparativa entre el N-R y AI de la red entrenada [3, 4, 2000, 0.00001]. ....	144

## **CAPÍTULO I: INTRODUCCIÓN**

En el presente capítulo se explica a detalle el planteamiento del problema, la pregunta de investigación, así como los objetivos tanto generales como específicos, la justificación de la investigación, así como las delimitaciones y los antecedentes necesarios para comprender el contexto en el que se realiza el presente trabajo.

### **1.1 Antecedentes**

En la actualidad una de las ramas de las ciencias que cuentan con más demanda es la de la robótica, este término tiene muchas definiciones en ocasiones determinadas por el contexto en el que se utiliza, como por ejemplo el caso de Torres Vargas que dice que la robótica es:

“Una rama de la tecnología, que estudia el diseño y construcción de máquinas capaces de desempeñar tareas repetitivas o peligrosas para el ser humano. Las ciencias y tecnologías de las que deriva podrían ser: las diferentes ramas de la matemática, los autómatas programables, las máquinas automatizadas, la física principalmente mecánica y electricidad, la electrónica y la informática, entre otras.”

(Torres Vargas, L. Z., 2012, página 12, párrafo 8).

Si bien, los robots son máquinas que ayudan a realizar trabajos que pudiesen ser peligrosos para los seres humanos, estas máquinas deben seguir ciertas reglas, llamadas leyes de la robótica, donde los creadores de dicha maquinaria deben apegarse a estas, siendo estas leyes las siguientes:

“1. Un robot no debe dañar a un ser humano ni, por su inacción, dejar que un ser humano sufra daño.

2. Un robot debe obedecer las órdenes que le son dadas por un ser humano, excepto si éstas entran en conflicto con la primera ley.

3. Un robot debe proteger su propia existencia, a menos que ésta entre en conflicto con las dos primeras leyes.

4. Un robot podrá tomar el trabajo de un ser humano, pero no debe dejar a esta persona sin empleo.”

(Kumar Saha, S., 2011, página 2, párrafo: 3)

Lo anterior se definió ya que hasta la fecha los robots suelen producir cierto temor hacia los seres humanos, tanto porque se cree que los reemplazarán en sus trabajos como la idea de que en algún punto de la historia dominarán al mundo. Con lo anterior ciertamente se descartan las ideas anteriores.

Por otro lado, la robótica no podría existir sin los sistemas ciber físicos, los cuales hacen posible que los robots o maquinaria se enlacen con computadoras las cuales son

intervenidas por los operadores, estos sistemas a su vez se consideran aparatos inteligentes pues si bien existe una interacción entre el humano y las máquinas estas pueden ser entrenadas para funcionar por sí solas.

Además de los sistemas ciber físicos, otro aspecto importante en la cuestión de la robótica es la manufactura 4.0, que hace referencia al uso de las tecnologías más avanzadas hasta la fecha, esto a su vez engloba conocimientos como son el internet de las cosas, sistemas integrados o los datos masivos, solo por mencionar algunos. En los sectores donde se ven avances significativos es en el sector de la aeronáutica y automovilístico, esto se debe al incremento en su producción, esto a su vez, impacta en otros sectores pues los avances tecnológicos tienen la ventaja de que se pueden adaptar a los distintos campos con gran facilidad, siendo así los sistemas de este tipo de manufactura los mejores y más aplicables hasta la fecha en la industria actual.

Uno de los tipos de robots más utilizados en la industria, son los robots paralelos, esto pues a pesar de que una de sus desventajas más grandes sea un limitado espacio de trabajo, estos suelen ser dinámicos siendo sus principales tareas en las fábricas el de tomar y colocar, así como el de ser centros de mecanizado a altas velocidades.

Según Díaz *et al.* (2018), un robot paralelo “consiste en una plataforma móvil unida a una plataforma fija mediante una serie de cadenas cinemáticas llamadas piernas”. Partiendo de esta definición, se puede hacer referencia a la primera aplicación industrial en 1942

presentada por Willard L.V. Pollard en 1940 la cual consistía en una máquina que pintaba vehículos con aerosol de forma automática (Ver figura 1).

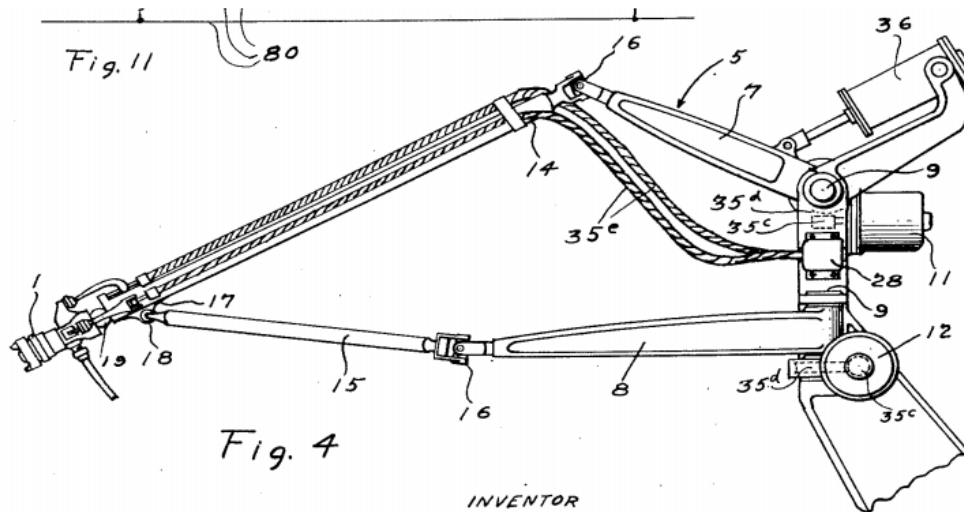


Figura 1. Primera aplicación industrial de los Robots Paralelos patentada en 1942. (Díaz, 2018)

A partir de este robot, comenzaron a crearse otros más, hasta llegar al llamado mecanismo Stewart el cual es utilizado bastante por los astronautas y pilotos para realizar entrenamientos y simulaciones de vuelo, como se ve en la figura 2.



Figura 2. Plataforma Stewart usada por Lufthansa como simulador de vuelo (Díaz, 2018).

Existen muchos tipos de robots utilizados en el ámbito de tomar y colocar algunos de los más populares son los robots tipo SCARA los cuales “permiten posicionar el elemento terminal, y por ende la pieza, en el espacio cartesiano, además también puede realizar una rotación” (Díaz, 2018). También los robots DELTA son populares pues a diferencia de los anteriores es que requiere menos masa e inercia para soportar las cargas que se le presentan (Ver figura 3).



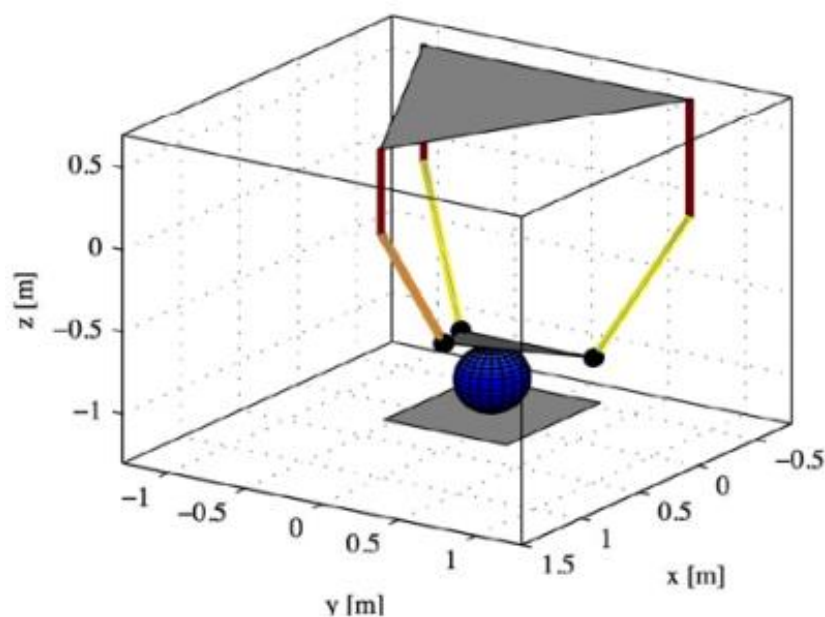


Figura 4. Esquema de funcionamiento de un robot híbrido donde el robot paralelo provee de 3 ejes (Díaz, 2018).

Un ejemplo de que se pueden combinar tanto los robots paralelos como la inteligencia artificial es la investigación realizada por Jiménez et al, (2021a), donde mediante un modelado de cinemática inversa de tres grados de libertad usando cuaterniones e inteligencia artificial logran resolver el problema de cinemática inversa. La estructura de la red neuronal resultó en tres capas ocultas con 25 neuronas por capa, así como una capa de salida. El problema cinemático inverso se modela como un sistema de seis ecuaciones no lineales con seis incógnitas, para luego de que estas fuesen probadas usar los resultados para crear una comparación entre el método matemático de Newton Raphson y el entrenamiento neuronal.

Por otro lado, en otra investigación realizada por Jiménez et al, (2021b ) estudiaron el modelado en dos configuraciones de un mecanismo paralelo plano de 5R y 2 grados de libertad, así como la solución del modelo de cinemática inversa usando inteligencia artificial, en este caso se usó un nuevo método de modelado cinemático el cual analiza 2 de las configuraciones del sistema rotacional en relación con las dos estructuras, esto mediante modelos matemáticos desarrollados con números complejos, el problema cinemático inverso por su parte se define por ocho ecuaciones no lineales con sus 8 incógnitas resuelto mediante el método de Newton Raphson. Por su lado, la red neuronal se entrenó a partir de una base de datos creada respecto a la configuración de los mecanismos incluyendo posiciones singulares; para el entrenamiento de la red se utilizó el algoritmo Levenberg – Marquardt, y al final los autores usaron una trayectoria lineal interpolado para entender mejor la eficiencia del entrenamiento.

Como se puede ver, los campos de la ciencia, en especial de la robótica, pueden mezclarse para generar soluciones más eficientes sin necesidad de que dichos campos sean enemigos, el ejemplo que se dieron anteriormente son la prueba de ello pues se mezcla robótica, inteligencia artificial y problemas cinemáticos, si la ciencia continúa a ese ritmo, muchos de los problemas actuales podrían ser solucionadas en lapsos de tiempo cada vez más cortos.

## 1.2 Planteamiento del problema

La Inteligencia Artificial es un campo de aplicaciones de la ingeniería de alta prioridad para el progreso tecnológico, debido a que es un área en donde se sustenta la manufactura 4.0

o fábrica inteligente. A pesar de que existen diversas aplicaciones de la Inteligencia Artificial en muchos campos del conocimiento, en el área de la ingeniería mecánica, y en especial en las aplicaciones de la misma en la robótica, sigue faltando modelado y simulación inteligente. A pesar de los enormes beneficios que se obtienen al usar métodos y procedimientos de la Inteligencia Artificial en lugar de los métodos los numéricos y analíticos tradicionales, la Ingeniería Mecánica se ha quedado atrasada, por lo que es necesario crear conocimiento en esta área.

Por otro lado, la mecatrónica del futuro estará basada en los llamados sistemas ciber-físicos, por lo que es necesario estudiar y analizar dichos sistemas, en especial la robótica. Los sistemas ciber-físicos son la base de la cuarta revolución industrial y dentro de su constitución se encuentran sistemas inteligentes, de allí la importancia de masificar más aplicaciones de la Inteligencia Artificial, sobre todo en campos como la Ingeniería Mecánica.

En este sentido la robótica paralela resulta ser un área de especial interés para la mecatrónica y para la mecánica, en particular en la mecánica del cuerpo rígido. El modelado cinemático de este tipo de robot es clave para estudiar sus movimientos. Sin embargo, los robots paralelos, al ser estructuras complejas, requieren de un tratamiento de modelado especial que en general resulta ser no lineal, particularmente los modelos con los que se formula el problema cinemático directo. Esta característica de no linealidad requiere que los modelos matemáticos deban ser resueltos con métodos numéricos. Estos métodos requieren un alto procesamiento computacional y se requiere programación robusta para

implementarlos. Los algoritmos de la Inteligencia Artificial, como es el caso de las redes neuronales, se han vuelto una opción para resolver los problemas no lineales en la robótica.

### 1.3 Pregunta de investigación

¿Qué beneficios se obtendrían al entrenar una red neuronal que pudiera resolver el problema cinemático inverso asociado a un robot paralelo delta de dos grados de libertad?

### 1.4 Objetivos

#### 1.4.1 Objetivo general

Seleccionar la estructura de red más adecuada y entrenar una “red neuronal especial” para solucionar el problema cinemático inverso de un robot paralelo planar de dos grados de libertad, utilizando el Algoritmo de Entrenamiento de Retropropagación de Regularización Bayesiana y el álgebra de los números complejos, para aplicaciones en robótica paralela y en sistemas mecatrónicos.

#### 1.4.2 Objetivos específicos

- Estudiar el modelado cinemático de posición del robot en estudio.
- Formular el problema inverso de posición
- Generar una base de datos de configuraciones del robot con el uso del problema inverso y el método de Newton-Rapshon
- Seleccionar una configuración de red neuronal para el entrenamiento

- Entrenar red neuronal
- Probar los resultados obtenidos

### 1.5 Justificación

El desarrollo de este trabajo de tesis tendrá como principal beneficiaria a la Ingeniería Mecánica y a los sistemas mecatrónicos robóticos, ya que el modelo y la solución del problema cinemático inverso del robot en estudio será parte de las aplicaciones de la Inteligencia Artificial en la ingeniería Mecánica. La docencia en matemáticas y las materias en mecanismos y robótica que se imparten en la Universidad La Salle Noroeste se verían también beneficiadas con los resultados de este trabajo debido a que puede usarse como ejemplo de aplicación de las materias de cálculo y cálculo vectorial, así como la de métodos numéricos. Del mismo modo las materias que centran el estudio en la mecánica del cuerpo rígido, como es el caso de los mecanismos y la robótica, podrán usar este trabajo como base para enseñar cinemática aplicada a los cuerpos rígidos.

Finalmente, la presente tesis podrá ser utilizada por los alumnos de la carrera en Ingeniería Mecatrónica en el desarrollo de proyectos de finales de clase que requieran de un robot paralelo.

### 1.6 Delimitaciones

- Este trabajo de tesis toma el modelado del robot propuesto por Jiménez et al. (2021).

- El entrenamiento de la red solo considera un número máximo de 3 configuraciones o topologías.

## **CAPÍTULO II: MARCO TEÓRICO**

En el siguiente capítulo se expondrá la información más relevante que contiene el proyecto para llevarse a cabo su realización, tales como conceptos importantes como robótica, cinemática de robots, álgebra de números complejos, al igual que inteligencia artificial y el método de Newton-Raphson, ya que son necesarios para el desarrollo y solución del problema inverso y directo del robot. Además, se extenderán los temas con su debida descripción, abordando sus características y aspectos importantes

## 2.1 Robótica

La robótica en la actualidad es una de las tecnologías más utilizadas, sin embargo, su definición puede ser bastante amplia, puede considerarse como la rama tecnológica que estudia la construcción y diseño de máquinas que ayudan y facilitan tareas repetitivas y/o peligrosas para los humanos, así como también se puede hacer referencia al estudio, desarrollo y aplicación de los robots.

El término es atribuido a Isaac Asimov, además de que planteó las tres leyes o principios las cuales consisten en que un robot no puede lastimar ni permitir que se lastime a ningún ser humano, el robot debe obedecer a todas las ordenes de los humanos siempre y cuando respete la primera ley, y por último el robot debe autoprotegerse, al menos que para hacerlo entre en conflicto las leyes anteriores.

### 2.1.1 Definición de robot

La palabra robot deriva de la palabra del idioma checo *Robbota* o *Robotnik* que significa ciervo, esto haciendo referencia a los trabajos forzados, monótonos u obligados. Pero no fue hasta 1921 que Karel Capek autor checoslovaco combina ambas palabras para tener así el término Robot y usarlo así por primera vez.

A partir de la aportación de Kumar Saha en conjunto con la Organización Internacional para la Estandarización (ISO) Robot se define como “un manipulador multifuncional reprogramable, capaz de mover materiales, piezas, herramientas o dispositivos especiales,

a través de movimientos variables programados, para el desempeño de tareas diversas.” (Kumar, 2011).

Pero según Chong “los robots son dispositivos compuestos de sensores que reciben datos de entrada y que pueden estar conectados a la computadora” (Chong M, 2009).

Como se dijo anteriormente existen una infinidad de definiciones y conceptos alrededor de la palabra Robot, sin embargo, se puede concluir que se trata de un dispositivo controlado por una computadora la cual al dársele órdenes o instrucciones por el hombre esta las realiza para evitar accidentes o trabajos muy pesados para el hombre.

#### 2.1.2 Clasificación de robots

Un robot puede ser clasificado dependiendo de los criterios o características que este tiene, como por ejemplo su aplicación, su construcción, entre otras categorías, todo acatando las especificaciones que se busquen en una máquina específica.

#### 2.1.3 Aplicaciones de los robots

En la actualidad los robots están ganando terreno en una considerable cantidad de áreas desde el entretenimiento, el sector doméstico, militar e incluso en el alimenticio, las aplicaciones de los robots crecen día con día a un paso acelerado, pues muchos aparatos que parecían que solo se quedarían en la ciencia ficción hoy son realidad e incluso tomados con tanta naturalidad que ya son parte de la rutina diaria de todos.

##### 2.1.3.1 En la industria

Este sector fue de los primeros en donde se aplicó un robot, en especial para manejar materiales radioactivos, en los años 50 por los laboratorios en Norteamérica Argonne. En 1975 la industria automotriz comenzó a utilizarlos para la reducción de costos y fluidez en su producción.

Las industrias por lo general que fabrican de algún producto, suelen tener materiales o procesos que pueden ser un peligro para el ser humano, por lo que es importante conocer bien la tarea que se asignará a la máquina, en la mayoría de los casos se asignan trabajos como de soldadura, esto pues al estar manejando materiales de un peso mediano o alto para así con una precisión envidiable realizar su tarea, esta acción puede sonar algo sin importancia pero cuando de calidad y durabilidad se trata esta parte del proceso entra en juego, por lo que al ver la respuesta tan eficaz de las máquinas la demanda de los robots de soldadura ha incrementado considerablemente.

Otra aplicación en la industria es la aplicación de materiales, así como de adhesivos y selladores, estos se encargan de aplicar el material, adhesivo o sellador el cual se seca después de un tiempo, al igual que en el caso de los robots soldadores, estos se han vuelto realmente populares por la precisión y rapidez con la que completan su tarea.

Además de las tareas anteriores, hay otra actividad que puede resultar riesgosa es la cuestión del corte de materiales, al utilizar este tipo de máquinas se reduce en gran medida la cantidad de accidentes dentro de las empresas, así como obtener productos con cortes exactos y precisos en tiempos de producción muy pequeños.

Existen muchas más actividades que los robots industriales realizan día con día en las empresas, sin embargo, las anteriores solo son algunos ejemplos de sus casi innumerables aplicaciones.

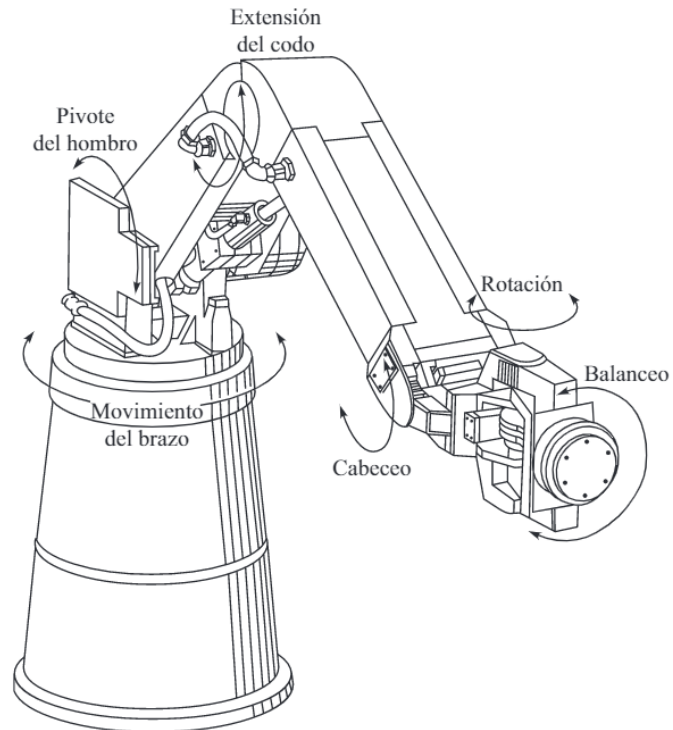


Figura 5. Robot industrial (Kumar Saha, 2011)

#### 2.1.3.2 En exploración espacial

El área de la exploración espacial es un sector relativamente nuevo pero muy importante, así como revolucionario, pues los descubrimientos realizados a través de las máquinas que se envían son esenciales para poder comprender algunos cuestionamientos actuales, así como resolver teorías y preguntas planteadas por científicos.

Un ejemplo de robots móviles, que son los más usuales en las exploraciones espaciales, es el Mars Rovers el cual es un vehículo autónomo todoterreno utilizado para explorar la superficie del planeta, deben ser autónomos puesto que controlarlos desde la Tierra consistiría un retraso considerable de alrededor de unos nueve minutos, desde que se envía la instrucción de movimiento hasta que se recibe la respuesta en Marte.

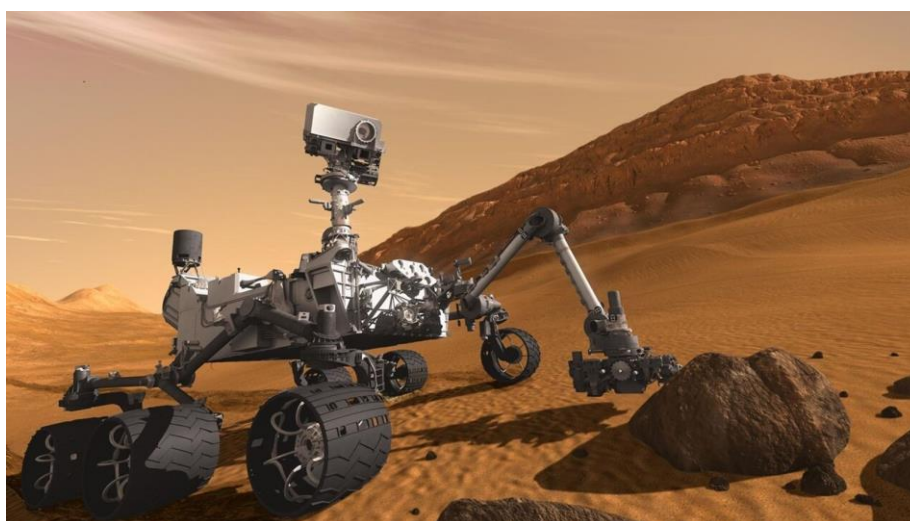


Figura 6. Mars Rovers. (National Geographic, 2021)

### 2.1.3.3 En educación

En los últimos años la tecnología ha evolucionado a un ritmo acelerado, se ha infiltrado en áreas que en décadas pasadas parecerían imposibles de imaginar si quiera, una de estas áreas es la educación.

Si bien el tema de integrar la tecnología a la educación, a pesar de ser un tópico bastante controversial se han realizado estudios donde la respuesta ha sido positiva puesto que conforme la sociedad se digitaliza, la educación lo hace de igual manera, tal como dice Vivas

Fernández en su artículo de investigación e Integración de la robótica en la educación primaria:

“En el ámbito educativo se pretende avanzar al ritmo que la sociedad evoluciona, donde debería darse el primer paso para que las futuras generaciones de ciudadanos aprendan a gestionar de una forma responsable, respetuosa con la diversidad y la multiculturalidad, tolerante y solidaria toda la tecnología a su alcance. Se trata de ofrecerles a los niños y niñas las estrategias necesarias para afrontar los retos tecnológicos y los problemas de la sociedad actual, integrando en su vida diaria los recursos más beneficiosos e innovadores.” (Vivas y Saes, 2019, Pág 121, Párrafo 1).

AL digitalizar la educación se estaría dando un paso enorme al comenzar a formar a los futuros innovadores e inventores dando bases sólidas desde sus primeros pasos.

#### 2.1.4 Robots paralelos

En la actualidad existen un sinnúmero de tipos de robots, así como de clasificaciones e incluso aplicaciones de estos, en este caso abordaremos lo que es un robot paralelo donde “un diseño modular consiste en un conjunto de módulos estandarizados, tales como actuadores, articulaciones, eslabones y efectores finales, los cuales pueden conectarse y desconectarse rápidamente para obtener diferentes configuraciones o grados de libertad en un robot manipulador” (Sánchez, González, y Balmaceda, 2015).

Este tipo de robots aún son un reto en los tiempos actuales puesto que aún no se sigue una metodología fija para poder hacer los estudios pertinentes para las configuraciones o geometrías convenientes para optimizar el desempeño del aparato mismo.

“La principal ventaja de los robots paralelos viene dada por la capacidad de distribuir las cargas aplicadas sobre el elemento terminal entre las piernas o cadenas cinemáticas abiertas que unen la plataforma móvil a la plataforma base.” (Díaz, 2018). Sin embargo, a pesar de que el peso se distribuye, tienden a ser aparatos muy voluminosos y pesados, limitando así el espacio de trabajo, además algunas configuraciones son algo complejas para su estudio.

#### 2.1.4.1 Diferencias entre robots paralelos

Principalmente las diferencias entre estos robots son principalmente por el tipo de movimiento donde se encuentran dos grandes grupos los robots planares y los robots espaciales.

El primer mecanismo está basado en el plano, limitándolo solo a 2 o 3 grados de libertad, lo cual hace referencia al movimiento de traslación en el plano, así como la de rotación sobre un eje perpendicular al mismo, pueden existir bastantes configuraciones con este tipo de mecanismos.

Los robots espaciales trabajan en todo el espacio tridimensional, pueden ser que cuenten con los tres grados de traslación o incluso que solo sean capaces de orientarse, y pueden contar con 2 o incluso 6 grados de libertad.

#### 2.1.4.2 Aplicaciones de los robots paralelos

Dado que las configuraciones de estos robots son muy versátiles, esto hace que el campo de aplicación sea igual de dinámico, abarcando desde micro robots hasta grandes robots

que soportan cargas muy pesadas, y sus aplicaciones pueden variar de campo desde cargar pesos muertos en industrias ensambladoras o incluso en el área médica hasta dispositivos voladores o experimentales.

Una de las primeras aplicaciones de los robots paralelos fue en los años 50, “el dispositivo fue propuesto para pintar vehículos de forma automática con pintura de aerosol y posteriormente fue patentado como dispositivo para controlar el posicionamiento de una herramienta.” (Díaz, 2018). Dando el inicio de la era de los robots paralelos.

#### 2.1.4.2.1 En la industria

Naturalmente en las empresas es común utilizar distintos tipos de máquinas, en el caso de los robots paralelos una de sus funciones es el de colocar y recoger, como su nombre lo indica el aparato toma una pieza para rotarla o simplemente moverla hacia otra plataforma para colocar en el lugar antes definido, estos pueden ser usados en empresas que empaquetan alimentos, cortes láser de alta precisión, al igual que mecanizados de piezas, entre otros.

Otra área industrial en la que este tipo de robot se aplica es en la del desarrollo de centros de mecanizado, donde las máquinas por sí solas crean cortes, por lo general, rotatorios, este tipo de robots suelen dejar una geometría y acabado superficial prácticamente impecable, siendo así una excelente alternativa pues su configuración puede ser muy variada.



Figura 7. Robot Quattro comercializado por Adept (M. Díaz-Rodríguez, 2018)

#### 2.1.4.2.2 En telescopios

En el ámbito espacial, los robots paralelos son utilizados para la posición de los pesados e inmensos telescopios, los cuales son los responsables de poder obtener imágenes impresionantes tanto de galaxias lejanas como de nuestro sistema solar, al tener distintas configuraciones y tolerar cargas con pesos elevados, los robots paralelos son la mejor opción para este tipo de tareas.



Figura 8. Plataforma de Gough (M. Díaz-Rodríguez, 2018)

#### 2.1.4.2.3 En simuladores de vuelo

En la industria aeroespacial, la simulación de vuelo es una tarea de suma importancia pues se reduce el índice de incidentes, por lo que se adaptaron los robots paralelos mediante una cabina sobre una de estas máquinas, la cual es controlada por el piloto.

El robot en sí es el encargado de poder sostener el peso de la cabina, así como de poder simular el vuelo de un avión, o en otras circunstancias más recreativas incluso de una nave espacial.

#### 2.1.4.2.4 En aplicaciones médicas

La medicina es un área muy importante y delicada, la cual necesita de precisión y eficacia para así poder salvar vidas, y al igual que en todas las ciencias esta debe evolucionar, es aquí donde entran los robots, en especial en las cirugías, hasta el momento el único robot comercial es el sistema Da Vinci, el cual el médico controla mediante una computadora los movimientos del robot al momento de hacer la operación.

En la actualidad existen otros robots paralelos aún en la etapa de prototipado, los cuales ayudarán en cirugías de espina dorsal, cirugía de ojos, o incluso hacer varias operaciones laparoscópicas, aunque aún necesitan ser aprobados para poder ser utilizados en los hospitales del mundo.



Figura 9. Robot paralelo P-arm utilizado en cirugía laparoscópica (M. Díaz-Rodríguez, 2018)

## 2.2 Cinemática de los robots

La cinemática es una derivación del estudio del movimiento que forma parte de la rama de la Física. El término viene de la palabra griega kineo que significa movimiento, se encarga de estudiar el movimiento de los cuerpos, ignorando de donde procede el movimiento, es decir, solo estudia la trayectoria del cuerpo en función del espacio y el tiempo.

Según Torres (2012) para aplicar la cinemática en un robot debe conocerse su trayectoria y la configuración del sistema. La cinemática:

“Se interesa por la descripción analítica del movimiento espacial del mismo como una función del tiempo, y en particular, por las relaciones entre la posición y la orientación de la herramienta del robot con los valores que toman sus coordenadas de las articulaciones.” (Torres, 2012, pág. 92, párrafo 4).

Además, existen dos tipos de cinemática, la directa y la inversa, según Ramírez y Rubiano (2012), “se aborda de forma directa para así calcular la posición del punto final del robot como función de los valores articulares (ángulos) y de forma inversa para calcular el valor de las coordenadas articulares como función de la posición final.”

Dado que la cinemática inversa es mucho más compleja que la directa, existen diversos métodos para solucionarla, por lo general se utilizan cálculos los cuales pueden proporcionar distintas configuraciones a partir de una sola ecuación, por lo que es común utilizar algoritmos para acercarse a una sola solución numérica.

Otra alternativa es el uso de redes neuronales, esta opción puede ser la más rápida y precisa, pero para el entrenamiento de la red es necesario usar la cinemática directa.

### 2.2.1 Cadenas cinemáticas

Cano (2017) explica que cuando se afirma “que, si una serie de segmentos superpuestos estuviesen conectados mediante unas juntas, se crearía un sistema que permitiría un movimiento completo de un segmento, que influiría en otro segmento dentro del mismo sistema cinético”. Además, se debe tener en cuenta que estos sistemas deben cumplir las características de construcción sencilla, fiabilidad y que son de menor costo.

Existen dos tipos de cadenas cinemáticas, las abiertas que consisten en que la barra solo se conecta a una sola barra, mientras que las cerradas son cuando están conectadas como mínimo a otras dos barras. La configuración de estas cadenas se da a partir del número de barras y pares que estas contienen.

### 2.2.2 Hipótesis de cuerpo rígido

Primero se define que es un cuerpo rígido y esto lo responde Bastián et al. (2007): “un cuerpo rígido es aquél en el que la distancia entre cualquier par de puntos permanece constante, es decir, es un cuerpo ideal cuyas dimensiones no cambian en ninguna circunstancia”.

El cuerpo rígido usualmente es estudiado por las leyes de Newton las cuales son tres, el principio de la inercia, el principio fundamental de la dinámica y la ley de la acción y reacción.

La primera hace referencia a que todo cuerpo está en reposo o movimiento uniforme rectilíneo siempre y cuando no exista una fuerza que actúe sobre este, es decir, que si el cuerpo está en movimiento uniforme y rectilíneo este seguirá hasta que algo externo ejerza una fuerza en este o se detenga.

La segunda dice que “Si sobre un cuerpo actúa una fuerza, se produce un cambio en el movimiento, proporcional a la intensidad de la fuerza y en la misma dirección de ésta.” (Figuroa, 2010). En esa ley se habla puramente de intensidad, es decir, de vectores y es aquí donde nace la ecuación que dice que la fuerza es igual a la masa por aceleración, y la unidad de fuerza que se utiliza es el Newton o la Dina, donde un Newton equivale a cien mil Dinias.

La tercera explica que “Cuando un cuerpo ejerce una fuerza (acción) sobre otro, éste ejerce a su vez sobre el primero (reacción) una fuerza de igual intensidad y en sentido contrario” (Figuroa M., 2010). En resumen, la acción y su reacción son iguales, pero en sentido opuesto. Esto a su vez dio lugar a la ecuación del peso el cual es igual a la masa por gravedad, donde la gravedad tiene un valor generalizado de 9.8 metros sobre segundos cuadrados.

### 2.2.3 Trayectoria

Para delimitar el movimiento de un cuerpo es necesario fijar un punto de referencia, ya sea que esté en estado de reposo o en movimiento, esto es fundamental al momento de poder definir o estudiar la trayectoria de un cuerpo. Así que la trayectoria es una línea geométrica que un cuerpo describe con su movimiento, la cual es compuesta por posiciones sucesivas

en un vector a lo largo del tiempo. Lo anterior da a su vez la ecuación de posición o trayectoria la cual representa el vector de posición en función al tiempo, se escribe en coordenadas cartesianas y en tres dimensiones lo que da:

$$\vec{r}(t) = x(t)\vec{i} + y(t)\vec{j} + z(t)\vec{k}$$

Donde  $r(t)$  es la ecuación de posición o trayectoria;  $x(t)$ ,  $y(t)$  y  $z(t)$  son las coordenadas con respecto al tiempo. Así como  $\vec{i}, \vec{j}, \vec{k}$  que representan los vectores unitarios en los ejes  $x, y, z$  respectivamente.

#### 2.2.4 Interpolación

Interpolación hace referencia a encontrar un valor el cual se encuentre entre dos o más puntos base conocidos, los cuales se aproximan por medio de polinomios. Existen dos tipos de interpolación la que es con espacios equidistantes y con espacios no equidistantes.

El primer tipo con espacios equidistantes, también conocida como interpolación de Newton, el cual su propósito es obtener el polinomio interpolado, pero para esto se necesita una ecuación base parecida a la siguiente:

$$P_n(x) = a_0 + (x - x_0)a_1 + (x - x_0)(x - x_1)a_2 + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})a_n$$

A su vez el algoritmo propone una regla para conseguir los coeficientes  $a_0, a_1, a_n$ . Siempre y cuando el polinomio interpolador cruce los puntos de interpolación lo que se obtiene:

$$P_n(x_0) = a_0 = f(x_0)$$

$$P_n(x_1) = a_0 + (x_1 - x_0)a_1 = f(x_1)$$

$$P_n(x_2) = a_0 + (x_2 - x_0)a_1 + (x_2 - x_0)(x_2 - x_1)a_2 = f(x_2)$$

⋮

$$P_n(x_n) = a_0 + (x_n - x_0)a_1 + \dots + (x_n - x_0) \dots (x_n - x_{n-1})a_n = f(x_n)$$

Como se observa  $a_0$  está ligada a  $x_0$ , mientras que  $a_1$  a  $x_0$  y  $x_1$  respectiva y sucesivamente, tomando en cuenta lo anterior se obtiene una nueva afirmación.

Teniendo lo anterior se puede decir que restando las dos primeras ecuaciones se tendría:

$$a_0 \equiv f[x_0], a_1 \equiv f[x_0, x_1]$$

$$a_1 \equiv f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Haciendo lo mismo con la segunda y la tercera se obtiene

Para así demostrar y afirmar lo siguiente

$$a_2 \equiv f[x_0, x_1, x_2] = \frac{f(x_2) - f(x_1) - f[x_0, x_1](x_2 - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0}$$

Suponiendo que se requiere conocer la expresión para  $a_{n-1}$  por lo que se construye el polinomio de grado  $n$ ,  $Q_n(x)$ , que se define de la siguiente manera.

$$a_n \equiv f[x_0, x_1, \dots, x_n] = \frac{f[x_1, \dots, x_n] - f[x_0, \dots, x_{n-1}]}{x_n - x_0}$$

$$Q_n(x) = P'_{n-1}(x) + \frac{x - x_n}{x_n - x_0} [P'_{n-1}(x) - P_{n-1}(x)]$$

Donde realmente  $P'_{n-1}$  es el polinomio interpolador que va de  $x_1$  y puede llegar hasta  $x_n$ . Al definir  $Q_n(x)$ <sup>1</sup> para  $x, x_2, \dots, x_n$ ,  $Q_n(x_i)$  este es igual al polinomio interpolador ( $P'_{n-1}(x_i)$ ) y a su vez es igual a  $f(x_i)$ , ya que  $P'_{n-1}(x)$  menos  $P_{n-1}(x)$  invalida estos puntos. Lo mismo sucede en  $x_n$  que dice que  $Q(x_n)$  es igual a  $P'_{n-1}(x_n)$  y esto es igual a  $f(x_n)$ . En  $x_0$  se cumple la siguiente expresión

$$Q_n(x_0) = P'_{n-1}(x_0) - [P'_{n-1}(x_0) - P_{n-1}(x_0)] = P_{n-1}(x_0) = f(x_0)$$

Posteriormente,  $Q_n(x)$  concuerda con  $P_n(x)$  esto porque el polinomio interpolador es único, esto hace que el coeficiente en  $x^n$  es, por ende,  $a_n$ . Como ya se dijo la fórmula es válida para  $P_n(x)$  y  $P'_{n-1}(x)$ , entonces se pueden identificar las potencias en  $x^n$  en ambas partes, donde el coeficiente en  $x^n$  de  $Q_n(x)$  es:

Conociendo la ecuación anterior se puede definir el polinomio interpolador como:

$$a_n \equiv f[x_0, x_1, \dots, x_n] = \frac{1}{x_n - x_0} (a'_{n-1} - a_{n-1}) = \frac{f[x_0, x_1, \dots, x_n] - f[x_0, x_1, \dots, x_{n-1}]}{x_n - x_0}$$

$$P_n(x) = f[x_0] + f[x_0, x_1](x - x_0) + \dots + f[x_0, x_1, \dots, x_n](x - x_0) \dots (x - x_{n-1})$$

Este método hace posible conocer los coeficientes del polinomio interpolador fácilmente en forma de tabla, como en el siguiente ejemplo:

$$\begin{array}{cccc}
 x_0 & f_0 & & \\
 & & f[x_0, x_1] & \\
 x_1 & f_1 & & f[x_0, x_1, x_2] \\
 & & f[x_1, x_2] & f[x_0, x_1, x_2, x_3] \\
 x_2 & f_2 & & f[x_1, x_2, x_3] \\
 & & f[x_2, x_3] & \\
 x_3 & f_3 & & 
 \end{array}$$

Este método es usado cuando se quiere evaluar en diferentes ocasiones el polinomio interpolador, además si se quiere aumentar el orden del polinomio interpolador los coeficientes de  $a_k$  que ya calculados no se modifican, a esto se le llama que es un valor de pertenencia.

Por otro lado, el método de interpolación con espacios no equidistantes también conocido como interpolación de Lagrange el cual al igual que el método anterior se encargan de encontrar un polinomio interpolador de grado  $n$  el cual tiene que pasar por  $n + 1$  puntos  $(x_i, y_i)$  de la siguiente forma

$$P_n(x) = \sum_{i=0}^n L_i(x)y_i$$

Donde las funciones  $L_i(i)$  cumplen con  $L_i(x_k) = 0$  siempre y cuando  $i \neq k$  y  $L_i(x_i) = 1$ . Esta última expresión asegura que se efectúe  $P_n(x_k) = y_k$ . Por otro lado las funciones tipo  $L_i(x)$  se forman de la siguiente manera:

$$L_i(x) = \frac{(x - x_1)(x - x_2) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_1)(x_i - x_2) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}$$

En su forma general se puede observar que  $L_i(x_i) = 1$  mientras que  $L_i(x_k) = 0$  mientras  $i \neq k$ . Sabiendo lo anterior se puede resumir la expresión del polinomio interpolador de  $f(x)$  en los puntos  $x_0, x_1, \dots, x_n$  de la siguiente forma:

$$P_n(x) = \sum_{i=0}^n f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x - x_j)}{(x_i - x_j)}$$

Si bien ambas expresiones pueden ser utilizadas de igual manera, dos de sus formas más utilizadas son la lineal y la parabólica que se ven a continuación:

Lineal:

$$P(x) = \frac{(x - x_0)}{(x_1 - x_0)} f_1 + \frac{(x - x_1)}{(x_0 - x_1)} f_0$$

Parabólica:

$$P_2(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} f_0 + \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} f_1 + \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} f_2$$

Este método es utilizado más bien de manera educativa pues es sencillo de entender el concepto de interpolación y cómo puede aplicarse, sin embargo, en los casos donde se debe evaluar varias veces pues no se tienen los datos necesarios para cada grado, pero a pesar de eso si es posible realizar algunas demostraciones con él.

#### 2.2.5 Formulación de los problemas directo e inverso

La cinemática directa se encarga de establecer la posición y orientación del extremo final de un robot, con respecto a un sistema de coordenadas usado como referencia, siempre y cuando se conozcan los valores de las articulaciones, para resolver esta problemática se utiliza el algoritmo de Denavit-Hartenberg el cual mediante los ángulos de las articulaciones se puede conocer la posición del extremo del robot.

Primeramente, se realiza un diagrama de los eslabones y las articulaciones del robot, así como de los ángulos que estas pueden tomar. Luego se localizan los ejes  $Z$ , los cuales van en cada eje, es decir en la primera articulación se situaría  $Z_0$  y así sucesivamente. Posteriormente, se asigna el origen del sistema de la base, donde en cualquier punto del eje  $Z_0$ . Los ejes  $X_0$  e  $Y_0$  se colocarán de tal manera que se forma un sistema dextrógiro (Ver figura 10) con  $Z_0$ .

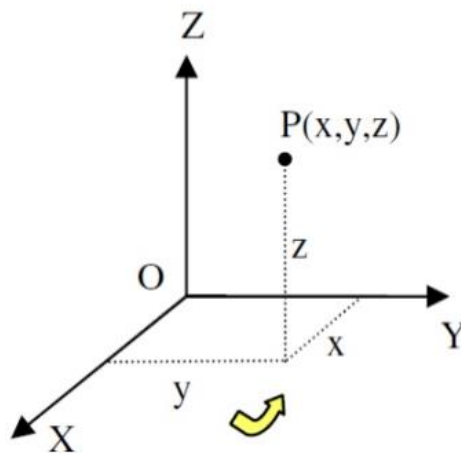


Figura 10. Sistema de Coordenadas Dextrógiro (s. a.,2016)

Después se sitúa el resto del sistema en la intersección del eje  $Z_i$  con la normal común a  $Z_{i-1}$  y  $Z_i$ , en caso de que los ejes fuesen a cortarse se agrega un punto de corte y si llegaran a ser paralelos se situarían en la articulación  $i+1$ . Para el eje  $X$  se coloca el  $X_i$  en dirección normal común a  $Z_{i-1}$  y  $Z_i$ , mientras que para el eje  $Y$  se posiciona  $Y_i$  de tal manera que se genere un sistema dextrógiro junto a  $Y_i$  y  $Z_i$  (Ver figura 11).

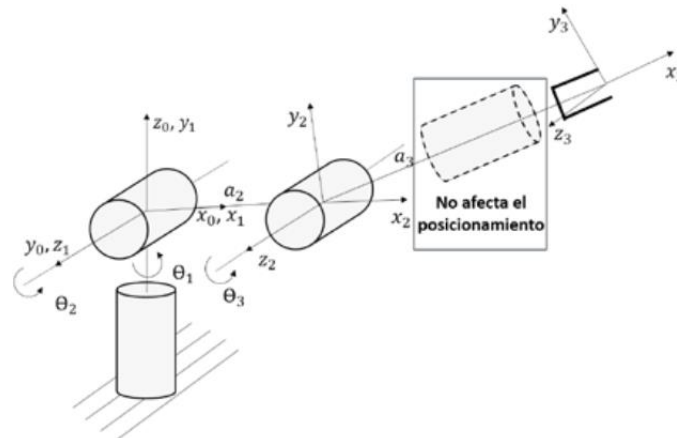


Figura 11. Representación simbólica y asignación de marcos de referencia (Cardoso, 2017).

Una vez realizado lo anterior se crea el sistema extremo del robot, donde se coloca el sistema de tal manera que  $Z_n$  concuerde con la dirección de  $Z_{n-1}$  y  $X_n$  sea normal a  $Z_{n-1}$  y  $Z_n$ . Luego se asignan los ángulos  $\theta$  donde cada  $\theta_i$  es el ángulo que va desde  $X_{i-1}$  hasta  $X_i$  girando sobre  $Z_i$ . Al igual que con los ángulos, se asignan las distancias  $d$  donde cada  $d_i$  hace referencia a la distancia desde el sistema XYZ i-1 hasta la intersección de las normales común de  $Z_{i-1}$  hasta  $Z_i$  a lo largo de  $Z_{n-1}$ . Mientras que las distancias  $a$  las cuales se definen como la distancia medida a lo largo de  $X_i$ , la cual coincide con  $X_{i-1}$ , la cual se desplazará de nuevo ( $S_{i-1}$ ) para que coincida con su origen ( $S_i$ ). Por otro lado, para obtener los ángulos  $\alpha$ , los cuales se obtienen al rotar  $Z_{i-1}$  para que llegue hasta  $Z_i$ , el primero se rota alrededor de  $X_i$ .

Después se deben obtener las matrices individuales de transformación A y, por último, se obtiene la matriz de transformación la cual relaciona el sistema de la base con el extremo del robot, para luego multiplicarlo por las matrices A.

Por otro lado, para el caso del problema inverso, se deben encontrar valores que adoptan las coordenadas de las articulaciones del robot  $q = [q_1, q_2, q_3, q_4, q_5, q_6]$  para que el extremo pueda posicionarse y orientarse según se desee para alcanzar una posición deseada en el espacio. Al igual que en el método directo se crea un diagrama y se asignan los grados, distancias  $a$  y ejes, dependiendo del problema que se presente.

La solución puede no ser única dependiendo del caso, pues pueden existir diferentes posiciones que solucionan el problema y orientan el extremo de la misma manera, sin embargo, para poder restringir un poco lo anterior se pueden crear algunas condiciones (Ver figura 12).

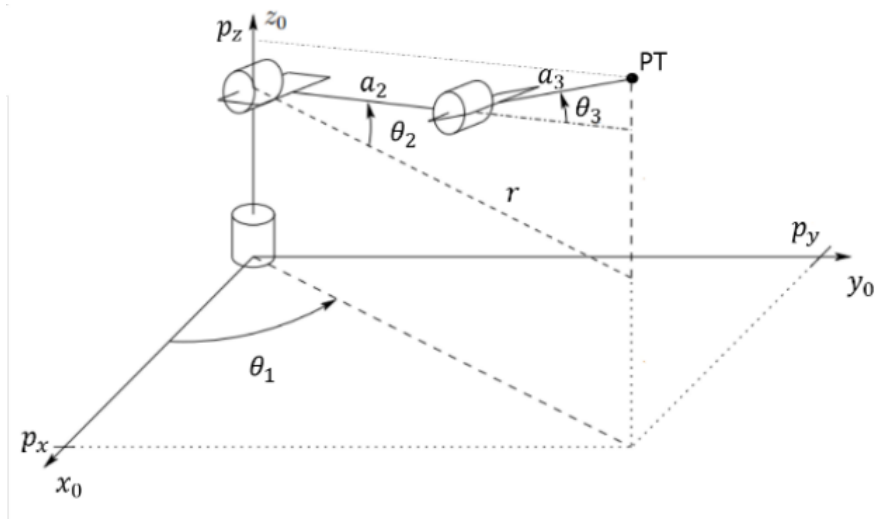


Figura 12. Representación geométrica. (Cardoso, 2017)

La solución para la posición, considerando un problema donde solo se toman los primeros tres ejes ( $q_1, q_2, q_3$ ) y que es dependiente del punto de espacio objetivo  $(x, y, z)$ , se utiliza:

$$q_k = f_k(x, y, z) \text{ para } k = [0,3].$$

Para solucionar el problema anterior, se sabe que se tiene un sistema de  $3 \times 3$ , teniendo las coordenadas del extremo del robot  $(x, y, z)$ , las coordenadas de las articulaciones  $(q_1, q_2, q_3)$  y las dimensiones físicas de los eslabones que serían  $(L, L_2, L_3)$ , con esto se puede decir que se tienen las coordenadas de partida  $(p_x, p_y, p_z)$  que sería la referencia donde se quiere posicionar el extremo del robot.

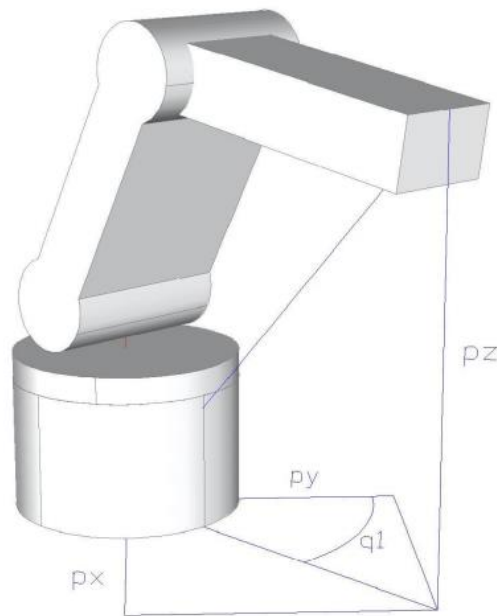


Figura 13. Esquemático de los 3 primeros grados de libertad del robot. (Rodríguez., s.f.)

Para obtener  $q_1$  se usa la expresión siguiente:

$$q_1 = \arctg\left(\frac{p_y}{p_x}\right)$$

Teniendo en cuenta que los eslabones 2 y 3 que están en un plano, se utiliza el teorema del coseno, con lo cual se obtienen las ecuaciones siguientes:

$$r^2 = p_x^2 + p_y^2$$

$$r^2 + p_z^2 = l_2^2 + l_3^2 + 2l_2l_3\cos(q_2)$$

$$\cos(q_3) = \frac{p_x^2 + p_y^2 + p_z^2 - l_2^2 - l_3^2}{2l_2l_3}$$

Con lo anterior se puede obtener  $q_3$  con respecto al vector de posición ( $\vec{p}$ ). Sin embargo, es mucho más sencillo usar la expresión del arcotangente en el lugar del arccoseno. Esto es porque:

$$\cos(q_3) = \pm\sqrt{1 - \cos^2(q_3)}$$

Se tendría:

$$q_3 = \arctg\left(\frac{\pm\sqrt{1 - \cos^2(q_3)}}{\cos(q_3)}\right)$$

Obteniendo así:

$$\cos(q_3) = \frac{p_x^2 + p_y^2 + p_z^2 - l_2^2 - l_3^2}{2l_2l_3}$$

Al ser una ecuación cuadrada, podrían existir dos posibles soluciones, ya sea negativa o positiva, resultando en las siguientes configuraciones del brazo (Ver figura 14).

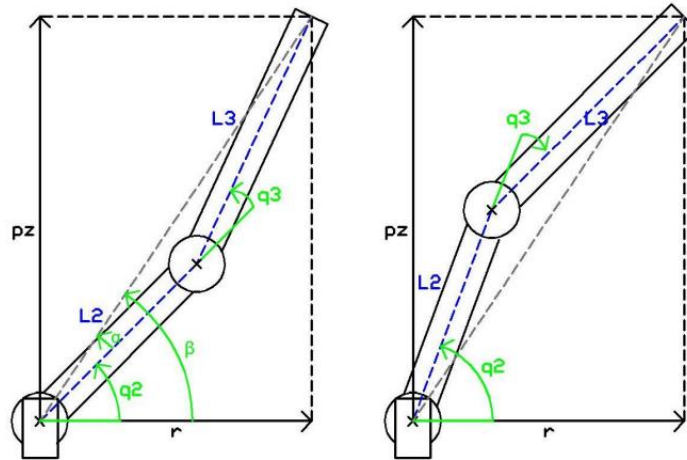


Figura 14. Configuraciones codo arriba y abajo. (Rodríguez, s.f)

El cálculo de  $q_2$  se obtiene a partir de  $\beta$  y  $\alpha$ :

$$q_2 = \beta - \alpha$$

Teniendo así

$$\beta = \arctg\left(\frac{p_z}{r}\right) = \arctg\left(\frac{p_z}{\pm\sqrt{p_x^2 + p_y^2}}\right)$$

$$\alpha = \arctg\left(\frac{l_3 \sin(q_3)}{l_2 + l_3 \cos(q_3)}\right)$$

Para finalmente despejar  $q_2$

$$q_2 = \arctg\left(\frac{p_z}{\pm\sqrt{p_x^2 + p_y^2}}\right) - \arctg\left(\frac{l_3 \sin(q_3)}{l_2 + l_3 \cos(q_3)}\right)$$

Con la ecuación anterior, se tendría de nuevo dos resultados uno positivo y otro negativo para  $q_2$ , las cuales corresponden a la posición del codo.

### 2.3 Álgebra de números complejos

Los números complejos son una combinación de números reales con símbolos, los cuales representan unidades imaginarias, además existen tres formas principales para este tipo de números, la binómica, polar o trigonométrica y exponencial o de Euler.

La forma binómica tiene una expresión muy característica, pues por lo general consta de un par de números reales y un número imaginario como se ve a continuación:

$$Z = a + bi$$

Donde  $a$  y  $b$  son números reales mientras que  $i$  hace referencia al número imaginario. Este tipo de números complejos es de las más utilizadas, además que se pueden realizar distintas operaciones con ellos, como suma, resta, multiplicación, división, potencia, por mencionar algunos.

Por otro lado, la forma polar, si bien puede partir de la forma binómica, se enfoca en el uso de coordenadas basándose de un número complejo de forma binómica, haciendo uso de las funciones trigonométricas, como en el caso siguiente donde para unas coordenadas  $(a, b)$  se utiliza:

$$\text{Sen } \theta = \frac{b}{r}$$

$$\text{Cos } \theta = \frac{a}{r}$$

Donde  $a$  se refiere al cateto adyacente,  $b$  el cateto opuesto y  $r$  la hipotenusa (ver figura 15).

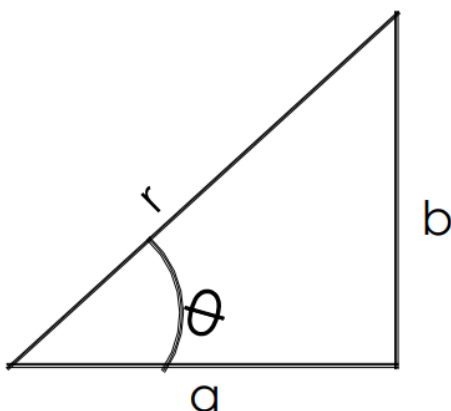


Figura 15. Definición de catetos e hipotenusa. (Ortiz, 2008).

Teniendo las expresiones anteriores se puede despejar  $a$  y  $b$  de tal manera, esto es:

$$a = r \text{Cos} \theta$$

$$b = r \text{Sen} \theta$$

Conociendo los valores de  $a$  y  $b$  estos se sustituyen en la primera ecuación, teniendo así:

$$Z = r \text{Cos} \theta + r \text{Sen} \theta i = r (\text{Cos} \theta + \text{Sen} \theta i)$$

Obteniendo así la forma trigonométrica, donde  $r$  es llamado el módulo mientras que  $\theta$  es conocido como el argumento principal que da grados entre  $0^\circ \leq \theta \leq 360$ . Al igual que con la forma binómica, en la forma polar también se pueden realizar algunas operaciones aritméticas.

Por último, la forma de exponencial, que se puede expresar de la siguiente manera:

$$e^{\theta i} = \text{Cos } \theta + \text{Sen}\theta i$$

Teniendo  $r$  y  $\theta$  el mismo significado en las expresiones de tipo polar, pero en este caso  $\theta$  está dado en radianes y no en grados teniendo así  $0 \leq \theta \leq 2\pi$ . Al igual que las otras formas de números complejos es posible realizar distintas operaciones aritméticas, pero se debe tener en cuenta que con la forma de Euler se tienen expresiones análogas en relación con las operaciones de forma polar.

### 2.3.1 Parejas ordenadas de números reales y propiedades de los números complejos

Una pareja ordenada hace referencia a dos objetos, símbolos, personas o cosas, que se mencionan en un orden específico, en el caso del álgebra se refiere a dos números ordenados de cierta forma, ya que si se mencionaran al revés se trataría de otra pareja completamente distinta. Su forma matemática habitual es la siguiente:

$$(x, y)$$

Donde  $x$  se refiere a la primera componente mientras que  $y$  es la segunda componente. Lo anterior suele utilizarse principalmente para localizar coordenadas en un plano cartesiano.

El producto cartesiano, de dos conjuntos  $A$  y  $B$ , da como resultado todos los posibles pares ordenados, los cuales se forman tomando un elemento del conjunto  $A$  y otro del conjunto  $B$ , y se escribe de la siguiente forma:

$$A \times B$$

Lo que daría a su vez:

$$A \times B = \{(x, y) \mid x \in A \quad y \in B\}$$

Lo anterior se refiere al producto cartesiano del conjunto  $A$  y  $B$ , donde  $x$  forma parte del conjunto  $A$ , mientras que  $y$  forma parte del conjunto  $B$ .

Por otro lado, los números complejos no solo se limitan a hacer operaciones, pues estos tienen propiedades las cuales son importantes resaltar.

La propiedad de Cierre para la suma, la cual se refiere a que si  $Z$  y  $W$  se sabe que son números complejos, entonces las operaciones  $Z + W$  y  $Z - W$  también serían números complejos. Lo mismo ocurre para la multiplicación donde se sabe que  $Z$  y  $W$  son número complejos, por lo tanto  $Z * W$  es un número complejo.

La propiedad Asociativa, que consiste en que si se sabe que  $Z$ ,  $W$  y  $U$  son números complejos, se tendría por ende lo siguiente:

$$Z + (W + U) = (Z + W) + U$$

Esta misma propiedad se cumple con la multiplicación donde se conoce que  $Z$ ,  $W$  y  $U$  son números complejos, se tendría:

$$Z * (W * U) = (Z * W) * U$$

Por otro lado, la propiedad Conmutativa dice que se tienen los números complejos  $Z$  y  $U$  entonces:

$$Z + U = U + Z$$

Esta misma propiedad se cumple con la multiplicación si se conoce que  $Z$  y  $U$  son números complejos, se tiene que:

$$Z * U = U * Z$$

Mientras que, la propiedad del Elemento neutro explica que el número complejo  $0 = 0 + 0i$ , es considerado el elemento neutro para la suma, se podría afirmar con la siguiente expresión  $Z = a + bi$ , obteniendo así:

$$Z + 0 = (a + bi) + (0 + 0i) = (a + 0) + (b + 0)i = a + bi = Z$$

Como se puede ver la ecuación  $0 = 0 + 0i$  realmente no modificó la expresión, probando que  $0 + Z = Z$ .

Para la multiplicación el elemento neutro se trata del número complejo 1, pues si se tiene el siguiente número complejo:  $Z = a + bi$  y se multiplica por 1 tendríamos:

$$Z * 1 = (a + bi) * 1 = (a * 1) + (b * 1)i = a + bi = Z$$

Al igual que con la suma, la expresión anterior no se alteró, de igual manera en que ocurre con  $1 * Z = Z$

La propiedad del opuesto dice que si se sabe que  $Z = a + bi$  es un número complejo, su opuesto sería por ende  $-Z = -a - bi$  que también es un número complejo, se puede ver que ambas expresiones satisfacen la propiedad de la siguiente manera:

$$Z + (-Z) = (-Z) + Z = 0$$

Para la multiplicación el inverso de  $Z$  se trata de  $Z^{-1}$  lo cual cumple con lo siguiente:

$$Z * Z^{-1} = Z^{-1} * Z = 1$$

La multiplicación cuenta con una propiedad extra la cual es la propiedad Distributiva, la cual explica que si se conoce que  $Z$  y  $W$  son números complejos, se pueden hacer las siguientes relaciones

$$Z * (W + U) = Z * W + Z * U$$

$$(Z + W) * U = Z * U + W * U$$

### 2.3.2 Caracterización de las rotaciones con números complejos.

Se puede decir que para expresar que se rota en un plano complejo, se debe suponer que el centro de rotación es  $a$  y el ángulo es  $\theta$ . La rotación se puede denominar  $R = R_{a,\theta}$ , primeramente, se puede tener una fórmula explícita para la rotación del ángulo  $\theta$  tomando en cuenta que se cuenta con un centro de rotación en 0. En este caso es mucho más conveniente y sencillo utilizar la forma exponencial de los números complejos.

Por lo que si se tuviese:

$$z = |z|e^{i\theta}e^{i\varphi} = |z|e^{i(\theta+\varphi)}$$

Una vez se define la expresión anterior, se puede llegar a la conclusión de que el radio vector del complejo  $e^{i\theta} * z$  es igual al radio vector de  $z$ , el cual es el que se rotó en un ángulo  $\theta$ .

Por lo que se consigue:

$$R_{0,\theta}(z) = e^{i\theta} * z$$

Lo cual se puede apreciar en la figura 16.

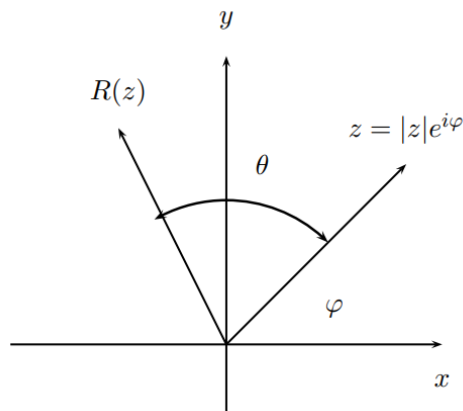


Figura 16. Rotación del ángulo  $\theta$  (Riviera, 2001).

Si se desea realizar una rotación del plano complejo en un ángulo  $\theta$  con un eje de rotación situado en cualquier punto  $a$ , entonces el centro de rotación se debe mover hasta  $0$ , luego se lleva a cabo la rotación con el mismo ángulo para por último, regresar hasta el punto  $a$  deseado, podemos llevar a cabo lo anterior por medio de la siguiente composición:

$$R_{a,\theta} = T_a \circ R_{0,\theta} \circ T_{-a}$$

Si se considera que  $z$  es un número complejo cualquiera, se sustituye en la expresión anterior obteniendo así:

$$\begin{aligned} R_{a,\theta}(z) &= T_a \circ R_{0,\theta}(T_{-a}(z)) \\ &= T_a \circ R_{0,\theta}(-a + z) \\ &= T_a \circ e^{i\theta}(-a + z) \\ &= e^{i\theta}(-a + z) + a \end{aligned}$$

$$= e^{i\theta} * z + (e^{-i\theta}(-a) + a)$$

En consecuencia, se deriva la expresión de rotación con centro en  $a$  y ángulo  $\theta$ , la cual puede darse por:

$$R_{a,\theta}(z) = e^{i\theta} * z + a(1 - e^{i\theta})$$

La cual puede ser entendida como que toda rotación con centro en  $a$  y ángulo  $\theta$ , se puede descomponer como una rotación del mismo ángulo, pero con un centro en 0, para luego moverlo al punto deseado.

También puede haber rotación de ejes, los cuales se puede dar iniciando con un número complejo cualquiera escrito de forma cartesiana, pues se compone de una combinación lineal entre complejos 1 e  $i$ :

$$z = a + bi$$

Al conjunto  $\{1, i\}$  anterior, se le denomina Base, la cual genera todos los números complejos. Pero si sobre un plano complejo se da una rotación  $R = R_{0,\theta}$ , teniendo un centro de rotación 0 y un ángulo  $\theta$ , entonces se transforman los elementos 1 e  $i$  en:

$$R(1) = e^{i\theta} * 1 = e^{i\theta} = v_1$$

$$R(i) = e^{i\theta} * i = e^{i\theta + \frac{\pi}{2}} = v_2$$

Teniendo así el plano mostrado en la figura 17.

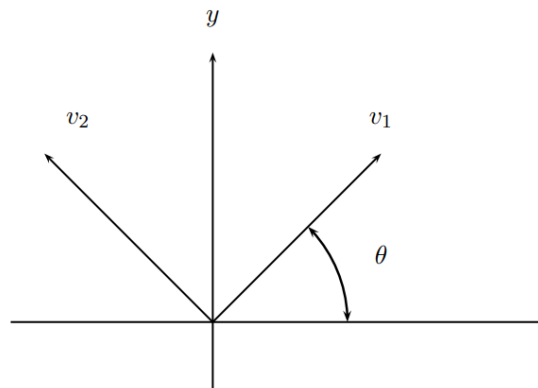


Figura 17. Rotación del plano. (Riviera F., 2001).

Considerando que  $z$  puede ser cualquier número complejo, se puede hacer una expresión como combinación lineal de  $v_1$  y  $v_2$ . Suponiendo que:

$$\begin{aligned} z &= c_1 v_1 + c_2 v_2 \\ &= c_1 e^{i\theta} + c_2 e^{i\theta + \frac{\pi}{2}} \end{aligned}$$

Para finalmente tener:

$$z * e^{-i\theta} = c_1 + c_2 i$$

De modo que bastaría con calcular la parte real junto a la parte imaginaria de  $z * e^{-i\theta}$  para poder encontrar los coeficientes de  $c_1$  y  $c_2$ .

#### 2.4 El método de Newton-Raphson

Es uno de los métodos para aproximar raíces, así como para la resolución de expresiones más utilizado, esto pues al solo necesitar un valor inicial que se acerque a la raíz deseada, lo hace sencillo de explicar y aplicar.

El valor inicial, puede obtenerse de muchas maneras, la más sencilla consiste en aproximar la función por la recta tangente al punto  $(x_0, f(x_0))$ , la cual se corta en el eje  $OX$  para luego despejar y encontrar el valor de  $x_1$ . El valor encontrado será una aproximación más acertada a la raíz de la función que el valor de  $x_0$ . Posteriormente, se continúa con el método realizando las iteraciones necesarias hasta encontrar un resultado dentro de la tolerancia.

Realizando este método por la serie de Taylor se desarrolla partiendo de la siguiente expresión:

$$f(x_{i+1}) = f(x_i) + f'(x_i)(x_{i+1} - x_i) + \frac{f''(\xi)}{2}(x_{i+1} - x_i)^2$$

Donde  $\xi$  hace referencia a cualquier punto en el intervalo  $x_i$  y  $x_{i+1}$ , luego de derivar la expresión anterior, se aplican las series de Taylor obteniendo así:

$$f(x_{i+1}) \cong f(x_i) + f'(x_i)(x_{i+1} - x_i)$$

Donde  $f'(x_i)$  es la primera derivada, así como la pendiente de la recta descrita. Pero en la intersección en el eje  $x$ ,  $f(x_{i+1})$  debe igualarse a cero, teniendo así:

$$0 \cong f(x_i) + f'(x_i)(x_{i+1} - x_i)$$

Despejando  $x_{i+1}$  se tiene:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Al ser un método geométrico esto hace que sea muy sencillo encontrar las aproximaciones de un punto a otro en una curva definida  $y = f(x)$  en la cual se trazan los puntos correspondientes a la aproximación (ver figura 18).

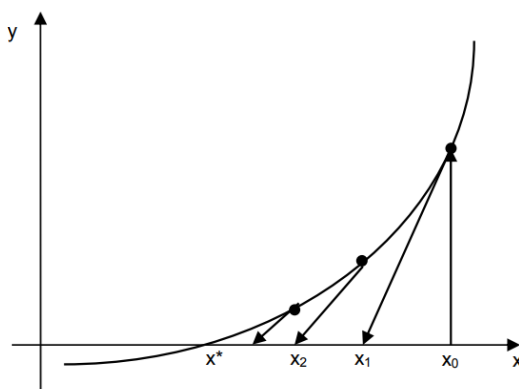


Figura 18. Método de Newton Raphson. (Chaillou, 2008).

Además, se deben tomar en cuenta las reglas de Fourier las cuales explica que si  $f: [a, b] \rightarrow \mathbb{R}$  es continua y dos veces continuamente diferenciable en  $[a, b]$ , entonces se puede verificar:

$$1. f(a)f(b) < 0$$

$$2. f'(x) \neq 0, \quad \forall x \in [a, b]$$

$$3. f''(x) \neq 0, \quad \forall x \in [a, b]$$

De tal manera que el método pueda converger si se toma  $x_0 = a$  o  $x_0 = b$  de modo que  $f'(x_0)f''(x_0)$ . Pero en caso de no ser así, se debe disminuir el intervalo o tomar un punto más cercano a la raíz, se acortará el intervalo hasta que la condición pueda cumplirse.

De igual forma, debe respetarse y seguir el algoritmo del método de Newton Raphson, donde para encontrar una raíz  $f(x) = 0$  se debe hacer lo siguiente:

Analizar la función  $f(x)$  donde se quiere conseguir la raíz y encontrar su derivada raíz  $f'(x)$ .

Encontrar o definir un punto de inicio  $x_0$ , de manera que  $f'(x) \neq 0$  para comenzar a hacer las iteraciones necesarias.

Si  $f'(x_0) = 0$ , significaría que se ha terminado el método, en caso contrario se debe seguir.

Si  $f(x_0) = 0$  quiere decir que la raíz en efecto es  $x_0$ , esto indica que el método terminó y se puede ir al paso 6. En caso contrario se debe hacer  $x_0$  igual a  $x_1$  o calcular  $x_1$  con la siguiente expresión:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Repetir a partir del paso 3 hasta que se cumpla la siguiente condición:  $|f(x_0)| \leq \delta$ , o incluso puede ser otro criterio considerando el error relativo y el número de iteraciones que se realicen.

Mostrar el resultado de  $x_0$ .

#### 2.4.1 Análisis para dos variables

El método de Newton – Raphson en ocasiones puede extenderse al grado de que existan más de una variable desconocida, el método en sí sigue siendo el mismo, aunque por el hecho de que existan más incógnitas este tiene ligeras modificaciones como se explica a continuación:

Si se toma un sistema con  $m$  ecuaciones no lineales con  $n$  incógnitas, se tendría algo parecido a lo siguiente:

$$f_1 = (x_1, x_2, x_3, \dots, x_n) = 0$$

$$f_2 = (x_1, x_2, x_3, \dots, x_n) = 0$$

$$\vdots$$

$$f_m = (x_1, x_2, x_3, \dots, x_n) = 0$$

Que a su vez puede expresarse en notación vectorial, teniendo así:

$$f(x) = 0$$

En donde se toma  $f(x)$  como un vector de dimensión  $m$  así como  $x$  se toma como un vector con dimensión  $n$ . Para solucionar un sistema de ecuaciones como el anterior es necesario obtener el vector  $x$  donde la función  $f(x)$  pueda ser aplicado en el vector nulo. Posteriormente se define a la raíz que se obtenga como  $\vec{\alpha}$ . En caso de tener una aproximación a la raíz  $x_r$  al desarrollar la serie de Taylor sobre la raíz y designar  $e_r = x_r - \vec{\alpha}$ , se puede componer de la siguiente manera:

$$f(x_r) = f(\vec{\alpha}) + \nabla f(\vec{\alpha}) * e_r + \dots$$

Si se están manejando expresiones de primer orden y considerando que se desconoce  $\vec{\alpha}$  se puede proponer la siguiente igualdad:

$$f(x_r) = \nabla f(x_r) * (x_r - x_{r+1})$$

La expresión anterior delimita un vector  $x_{r+1}$  el cual corresponde al valor más próximo a la raíz  $x_r$ . Hasta el momento se ha usado la igualación  $f(\vec{\alpha}) = 0$  y sustituido la raíz por  $x_{r+1}$  en la expresión  $e_r$ . Mientras tanto, el término  $\nabla f(x_r)$  en el caso anterior es considerado como una matriz de tipo  $m \times n$ , que se alude por  $J$ , la cual se distingue con el nombre de matriz Jacobiana de la función  $f(x)$ , y que se construye de la siguiente forma:

$$J = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$

Suponiendo que la matriz Jacobiana no es cuadrada, la izquierda de esta se multiplica por la matriz inversa de  $J$ , la cual se escribe como  $J^{-1}$ , que corresponde de igual manera a una matriz con dimensiones  $n \times m$ , teniendo la siguiente relación:

$$x_{r+1} = x_r - J^{-1} * f(x_r)$$

La cual hace referencia a la generalización del método de Newton- Raphson para  $n$  variables. Este vínculo proporciona un nexo de recurrencia a partir de un vector de partida  $x_0$ , la cual convergerá a la raíz siempre y cuando  $x_0$  se encuentre en la zona de convergencia también llamado atractor, además de considerar si existe la matriz inversa de la Jacobiana, por lo tanto, conviene expresarlo de la siguiente manera:

$$J * x_{r+1} = J * x_r - f(x_r)$$

Tomando en cuenta que  $J$  se toma como la matriz jacobiana la cual se calculó en el punto  $x_r$ . Las iteraciones consisten en que cada una resuelva un sistema de ecuaciones lineales y posteriormente usar la solución que se obtuvo para calcular así la nueva matriz de coeficientes al igual que el nuevo vector de términos independientes, el cual implica la evaluación de  $(n + 1)m$  funciones. Usualmente lo que se suele hacer es actualizar el valor de  $x_r$  por un valor  $\delta x_r$ , de tal manera que se genera la siguiente expresión:

$$x_{r+1} = x_r + \delta x_r$$

La función anterior se puede escribir como una ecuación para  $\delta x_r$  teniendo así:

$$J * \delta x_r = -f(x_r)$$

La cual se resuelve en cada iteración, actualizando el valor de  $x_r$ , para detenerse cuando se cumpla la siguiente condición:

$$||\delta x_r|| < \varepsilon$$

## 2.5 Inteligencia artificial

Inteligencia se puede entender como la capacidad de entender o conocer algo, así mismo también se le conoce como la potencia intelectual, se basa principalmente en la cantidad de conocimientos individuales al igual que con las habilidades de manipular y redefinir los conocimientos de forma adecuada basándose en los datos proporcionados como información inicial para la resolución de una problemática o situación. También podría definirse como las aptitudes para integrar y comprender lo aprendido para luego utilizarlo

al enfrentar situaciones insólitas. (Chong, M. (2009). Robótica e inteligencia artificial. Santa Fe, Argentina, Argentina: El Cid Editor | apuntes.)

Por otro lado, la palabra artificial se puede interpretar como algo que no fue hecho de forma natural, es decir, es un producto fabricado por el hombre.

Al momento de unir estas dos expresiones se crea otro término que es la “inteligencia artificial”, la cual es la disciplina que se enfoca en el estudio de las tecnologías las cuales mediante máquinas y programas informáticos son aptos para realizar tareas complejas con eficiencia y habilidades cercanas, o en algunos casos superiores, a las del mismo ser humano.

#### 2.5.1 Métodos de inteligencia artificial

La inteligencia artificial no funciona por sí misma, si no que utiliza algunas herramientas para la resolución de problemas, estas herramientas están afiliadas a distintas técnicas o métodos que son muy importantes en el área de la inteligencia artificial, entre las más básicas están: la robótica, las redes neuronales, algoritmos, reconocimiento de patrones, búsqueda de soluciones, solo por mencionar algunas. (Inteligencia Artificial. (s. f.). Repositorio digital de la Facultad de Ingeniería - UNAM.)

##### 2.5.1.1 Algoritmos genéticos

Los algoritmos son secuencias que deben seguirse para solucionar un determinado problema, deben ser precisos pues si se omite algo el resultado podría variar creando así problemas a futuro, además todo algoritmo debe tener definibilidad, esto hace referencia

a que no importa la cantidad de iteraciones que se realicen las secuencias el resultado siempre tiene que ser el mismo, por último un algoritmo debe ser finito, si el algoritmo es correcto este debe concluir luego de cierto número de iteraciones teniendo un resultado, en su mayoría, favorable, de esta forma sabremos que el proceso es satisfactorio.

Un algoritmo genético intenta recrear lo que sucede en con la selección natural, donde solo los más capaces sobreviven y evolucionan, este tipo de algoritmos trabajan con la representación de una solución a un problema y sobre esta se comienzan a aplicar operadores genéticos para así crear combinaciones y así que se reproduzcan, para medir la oportunidad de solución se establece una función de aptitud la cual clasifica y determina si es adecuada la solución propuesta. Pueden ser tomados como un proceso de optimización y clasificación donde las soluciones existentes puedan mejorarlas o incluso crear nuevas, las cuales pueden traer consigo mejoras, las cuales pasan por el filtro que en este caso sería la función de ajuste. Una de sus características más sobresalientes es que mediante este tipo de algoritmos las soluciones que tienen más desventajas se le castiga, mientras que las más eficientes y adecuadas se les premia de tal manera que las mejores soluciones son capaces

de tener una propagación más eficaz. (Trigo Aranda, V. (s. f.). Algoritmos. Autores científico-técnicos y académicos.)

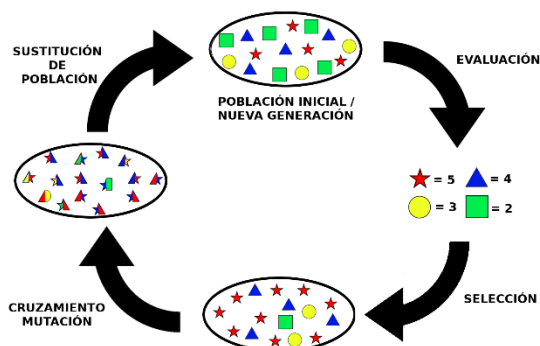


Figura 19. Proceso de evolución de los algoritmos genéticos. (Campos M. 2019)

#### 2.5.1.2 Redes neuronales

Las redes neuronales hacen referencia a la creación de sistemas los cuales están conformados por redes las cuales cuentan con un gran número de conexiones entre sus capas las cuales procesan la información, estas a su vez tienen funciones específicas, cada procesador de información debe estar aprendiendo constantemente mediante salidas y entradas de señales de entrenamiento.

Este tipo de sistemas son auto adaptativos, pues aprenden tanto de su entorno como del mismo código del que fueron creados, además de dinámicos pues como se dijo anteriormente estas van aprendiendo y con ello cambian constantemente sus resultados, en algunos casos incluso después de su etapa de entrenamiento.

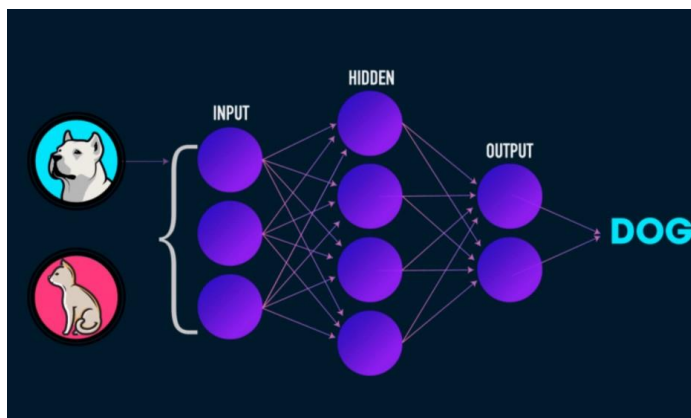


Figura 20. Red Neuronal Artificial. (Costa A., 2021)

### 2.5.1.3 Lógica difusa

La lógica difusa se entiende como la disciplina matemática que se encarga de resolver problemas utilizando más la certeza que la precisión como sería usual en la lógica clásica, a su vez tiene la aptitud de poder reproducir de una forma eficiente y aceptable el razonamiento humano, además suele ser muy flexible, tener una alta tolerancia a la imprecisión, así como la capacidad de modelar problemas no lineales y su fundamento de lenguaje es más de sentido común.

Si bien existen dos maneras de formar sistemas de este tipo, el primero llamado Mamdani donde sus salidas también forman parte de los conjuntos difusos, este método es utilizado cuando se quiere imitar de un modo más parecido a la que lo haría un humano, y el método Sugeno en el cual las salidas se toman como funciones constantes o lineales. (Ruvalcaba Coyaso, F. J. (2015). *Lógica difusa para la toma de decisiones y la selección de personal*. Universidad del Rosario.)

También debe tomarse en cuenta los criterios que se tomarán para seleccionar la mejor salida, algunos de ellos es el del método máximo, en donde el valor de salida elegido es el que donde la función característica del conjunto difuso es el mayor, puede no ser el mejor método esto porque varias salidas podrían alcanzar dicho valor. Por otro lado, el método del centroide selecciona el centro de gravedad de la función característica de salida, en este método solo se consigue una salida única. Por último, se tiene el método de altura en el cual se calculan los centros de gravedad de cada regla del conjunto difuso de salida para posteriormente calcular la media de todos los resultados obtenidos y así tener una salida.

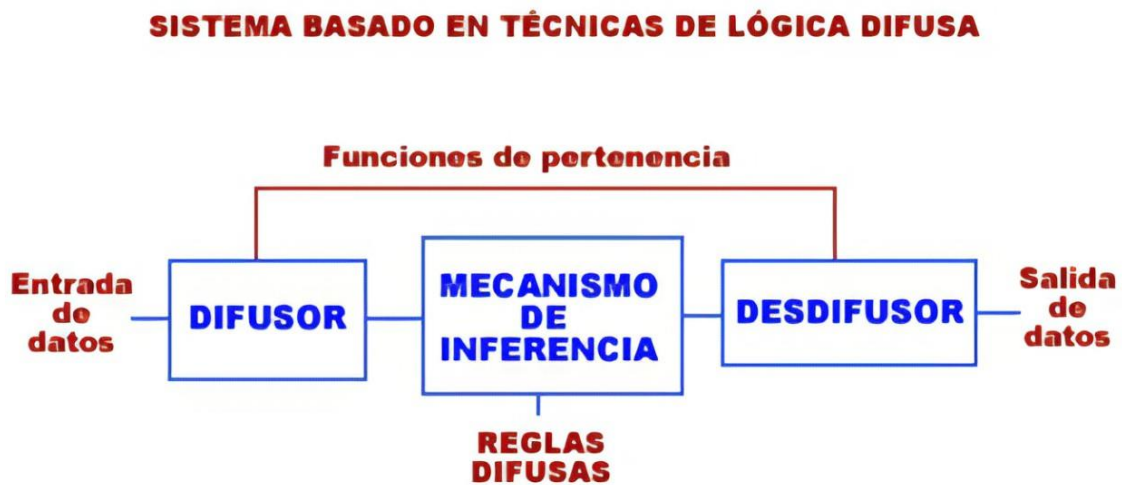


Figura 21. Sistema basado en lógica difusa. (Fronzino E. 2017)

#### 2.5.1.4 Otros métodos

La robótica es la disciplina que se encarga de estudiar, desarrollar y aplicar a los robots, los cuales hacen tareas repetitivas y muchas veces imposibles para el ser humano, es

importante que los robots que se desarrollen sean capaces de poder tanto recibir como emitir información, así como de entender lo que pasa a su alrededor, en resumen, las máquinas deben ser capaces de aumentar su eficacia en sus habilidades, así como en su autonomía.

El reconocimiento de patrones se trata de mediante distintas habilidades se puedan clasificar comportamientos o situaciones de cada subgrupo que se genere en cada grupo y mediante la asociación se obtienen diferentes conclusiones dependiendo del caso, en resumen, mide el parecido tanto de formas o comportamientos los cuales se repiten o aparecen constantemente.

Asimismo, la búsqueda de soluciones consiste como su nombre lo indica, en investigar las distintas maneras en que un problema puede resolverse, existe un tipo de búsqueda que puede facilitar esta tarea, se trata de la búsqueda heurística, la cual es una forma en la que se crean criterios o métodos para encontrar cual o cuales son las formas más eficaces para generar el resultado deseado.

La representación del conocimiento consiste en estructurar dicho conocimiento para que una computadora pueda entenderlo y a su vez trabajar con él, también se le conoce a este proceso como formalismo, este tipo de mecanismo ayuda a que el programa de Inteligencia Artificial pueda tomar decisiones de una forma más eficiente, así como planear, hacer reconocimiento de situaciones u objetos, así como otras funciones intelectuales. Para que una representación del conocimiento sea aceptable debe tener algunas características

como la de ser concisa, representar la información necesaria en su totalidad, exponer las restricciones propias del problema, hacer evidentes aspectos de relevancia, así como que el proceso sea lo más transparente posible.

Por último, otro de los métodos más utilizados para el desarrollo de sistemas inteligentes es el de procesamiento del lenguaje natural, es una de los protocolos más interesantes y complejos que posee la Inteligencia Artificial, esto pues consiste en traducir el lenguaje con el cual los humanos nos comunicamos, esto incluye tanto el hablado como el escrito, de tal forma que una computadora o máquina sea capaz de interpretarlo y por ende realizar alguna tarea.

#### 2.5.2 Aplicaciones de la inteligencia artificial

La Inteligencia Artificial está modificando la vida cotidiana de todos los seres humanos, siendo esta el principal sistema que se está desarrollando constantemente y que a su vez tiene un impacto notable en los avances tecnológicos, desde simples programas que predicen lo que el individuo desea escribir hasta maquinaria y robots que hacen decisiones por sí mismos, solo por mencionar algunos ejemplos de sus principales aplicaciones.

##### 2.5.2.1 En robótica

Los principales robots que contienen este tipo de protocolos son los robots espaciales, que si bien en muchas ocasiones son parcialmente autónomos, son prototipos que dan lugar a un rango de posibilidades mayor al momento de desarrollar redes de Inteligencia Artificial mucho más elaborados, un ejemplo de ello es el robot Curiosity lanzado en 2012 hacia

Marte, su principal misión fue examinar la superficie de Marte, su software además de detectar contornos, es capaz de analizar el brillo, forma, orientación y tamaño de lo que se encuentra a su alrededor, sin embargo el aspecto más interesante es que Curiosity es capaz de tomar decisiones por si solo acerca de qué objetivo es el que más pueda ser de utilidad a los científicos para así llevar a cabo su tarea de medir y tomar datos de los ejemplares del suelo.

Otro ejemplo de los robots espaciales es la sonda espacial Deep Space 1, la cual fue lanzada en 1998 donde su objetivo era corroborar el comportamiento de tecnologías en su entorno espacial, su software utilizaba búsqueda heurística esto para planificar las acciones que se tomarían a base de un sistema en tiempo real, sin contar que poseía un sistema que monitoreaba las acciones ejecutadas esto para evitar errores y en caso de que estos existiesen se pudieran corregir.

Otra área de robots con Inteligencia Artificial es la de los vehículos autónomos, quien inició esta tendencia fue la agencia DARPA en los años ochenta, comenzando con pruebas militares realizadas por universitarios, las pruebas no fueron tan satisfactorias por lo que eventualmente el programa se canceló, sin embargo dio lugar a toda una red de programas posteriores donde se obtuvieron resultados muy interesantes pues conforme el tiempo fue pasando la tecnología evolucionó, e incluso esta agencia inspiró a otras a crear sus propios vehículos tales son los casos como Amazon, Apple o Google, asimismo no solo se espera que sean coches los vehículos autónomos sino cualquier transporte. (Meseguer González,

P. y Ramon López de Mántaras Badia (2017). Inteligencia artificial. Madrid, Spain: Editorial CSIC Consejo Superior de Investigaciones Científicas.)

#### 2.5.2.2 En manufactura 4.0

En la actualidad se vive lo que se conoce como la cuarta revolución industrial, la cual como explica Salesforce (2018): “Es una manera de describir un conjunto de transformaciones en marcha y otras prontas a ocurrir en nuestra economía, sociedad y forma de vivir”.

La manufactura 4.0 es uno de los principales pilares de la revolución que se vive día a día, en especial porque es en este sector donde se ponen a prueba una gran cantidad de prototipos, además de que este tipo de manufactura actualizada está conformada tanto por partes virtuales como físicas, ambas siendo de suma importancia.

“Las piezas base son los productos inteligentes, caracterizados por disponer de electrónica, software embebido y conectividad. Se denominan sistemas ciber-físicos (CPS) y tienen capacidad de interactuar con otros sistemas y con humanos. El software les permite auto-gestionarse y tomar decisiones descentralizadas. Equipados con sensores captan información sobre su entorno, su uso y estado que pueden proporcionar a quien lo fabricó o gestiona su servicio. Sobre los CPS pueden ofrecerse servicios inteligentes y establecer nuevos modelos de negocio, aprovechando combinaciones innovadoras de servicios inteligentes para incrementar la creación de valor, aguas arriba o abajo de la cadena de valor.” (Del Val., J. L. (2016, 18 marzo). industria 4.0 | Revista Ingeniería. *REVISTA DEUSTO INGENIERÍA*.).

De acuerdo con Del Val, (2016), los sistemas de producción ciber – físicos son prácticamente la esencia de toda fábrica manufacturera que forma parte de la revolución 4.0, ya que son capaces de comunicarse, adaptarse, ser flexibles y personalizables, además mediante la digitalización de identificación de procesos, así como su automatización hace que las fallas

y soluciones se puedan identificar y resolver en tiempos mucho más cortos, además de minimizan el índice de error humano a prácticamente cero.

Con ayuda de la Inteligencia artificial se puede tener una cadena de suministro interconectada e inteligente, esto pues con los datos que las máquinas obtienen, estas por sí solas pueden hacer sus propias decisiones y aprender ciertos patrones necesarios para la resolución de problemas, generando así una cadena de procesos eficiente y competente contra los negocios globales.

### 2.5.3 Las redes neuronales

Las redes neuronales se pueden entender como sistemas o modelos tanto matemáticos como computacionales que, al ser inspirados directamente por la biología, donde cada neurona realiza tanto cálculo como procesamiento de información, esto hace posible que una computadora sea capaz de aprender y entender entre distintos patrones y formas, así como de crear soluciones complejas a distintos problemas determinados.

Cada red neuronal se conforma de un cierto número de neuronas conectadas entre sí, donde cada una de ellas posee una entrada y salida la cual a su vez hace posible que la función que tiene designada se realice lo mejor posible, y donde cada red debe estar sometida a un proceso de entrenamiento.

Algunas de las ventajas de una red neuronal es que además de que son capaces de adaptarse, pues realizan tareas luego de que fueron entrenadas o luego de que estas

consiguieron algo de experiencia realizando dicha tarea. Además, son capaces de auto organizarse pues la red neuronal es capaz de desarrollar un sistema de organización luego de que obtuvo su entrenamiento. También son capaces de tolerar ciertas fallas, si bien una red neuronal no está exenta de que la estructura pueda derrumbarse, si está teniendo la información suficiente podría seguir funcionando. Asimismo, este tipo de sistemas si se les adapta de cierta manera pueden realizar tareas en paralelo haciendo posible que se tenga una operación en tiempo real. Y, por último, teniendo las adaptaciones necesarias de hardware se puede tener un mejor rendimiento en las tareas de la misma red neuronal, haciendo que esta sea mucho más eficiente y óptima.

Una red neuronal se conforma de distintas capas, las principales son llamadas: capas de entrada, capas ocultas y capa de salida (Ver figura 22). Donde en la primera capa es donde el recibe la información ya sea de otro proceso o de otra red neuronal, dicha información pasa al segundo tipo de capa, la capa oculta quien se encarga de procesar la información recibida esta etapa de la red neuronal puede estar constituida por tantos niveles como sean necesarios así como conectarse de distintas formas creando así topologías variadas entre las neuronas para por último pasar a la capa de salida donde se transmite la información hacia el exterior.

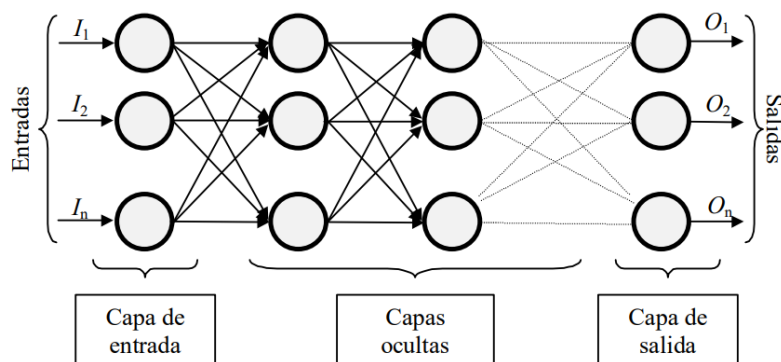


Figura 22. Ejemplo de una red neuronal totalmente conectada. (Matich, D. J. (2001, marzo). *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Universidad Tecnológica Nacional – Facultad Regional Rosario.).

Las neuronas a su vez se clasifican en dos tipos: binarias y naturales, donde las binarias toman valores dentro del intervalo  $\{0, 1\}$  o  $\{-1, 1\}$ , por otro lado, las neuronas naturales lo hacen dentro del rango  $[0,1]$  o  $[-1,1]$ , sin embargo, los pesos solo se limitan a cierto rango o intervalo en ocasiones específicas y mientras esto no ocurra no tienen alguna restricción.

La toma de decisiones se realiza como se ve en la figura 23, donde la salida es la decisión por tomar o la base para el proceso en general, mientras que los atributos son las variables que se deben cambiar para que la entrada al momento de que sea procesada pueda generar el resultado deseado.

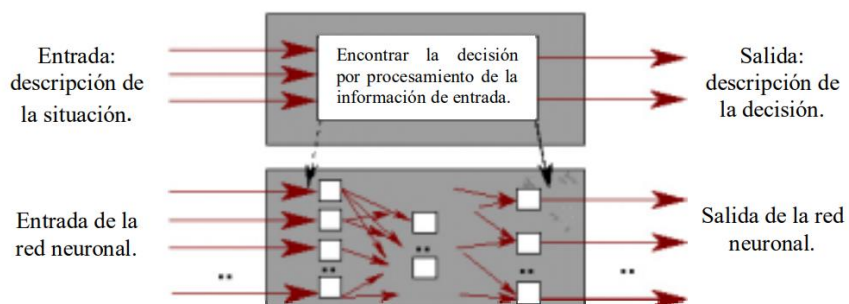


Figura 23. Esquema del proceso de decisión. Matich, D. J. (2001, marzo). Redes

Neuronales: Conceptos Básicos y Aplicaciones. Universidad Tecnológica Nacional –  
Facultad Regional Rosario.)

El aprendizaje es uno de los procesos más importantes de una red neuronal, esto consiste en crear y adecuar los cálculos para obtener la salida deseada, esto se realiza para cada conjunto de neuronas llamado constelación, vector o arreglo. Durante este proceso la topología y conexiones neuronales no pueden cambiar, sin embargo, los pesos que se encuentran en cada conexión si pueden variar generando así lo que se conoce como adaptación de pesos.

Existen dos tipos de aprendizaje, el aprendizaje supervisado y el no supervisado. El primer tipo de aprendizaje consiste en que, a partir de una variable externa al sistema, también conocido como supervisor o maestro, determina la solución que la red debe obtener partiendo de una entrada conocida, el maestro regula la salida de la red de tal manera que en los casos donde no coinciden la entrada con la salida se comienza un proceso donde los pesos de las conexiones comienzan a variar hasta que el maestro se acerque lo más posible a la salida que se desea.

Por otro lado, el aprendizaje no supervisado no tiene otros datos más que los de entrada, creando incluso un poco de ambigüedad en la salida, sin embargo, mediante categorización de características, regularidades o correlaciones la misma red debe ser capaz de organizar la información que se da en cada iteración para así poder compararla y obtener una salida deseada o que pueda funcionar para su futuro estudio.

Las redes neuronales a pesar de que son una tecnología relativamente nueva se utilizan en una gran cantidad de aplicaciones tanto militares como comerciales, dependiendo de su desarrollo pueden llegar a realizar tareas desde muy sencillas hasta realmente complejas, así como tener un rendimiento ejemplar con una gran tolerancia a fallas si se implementan con un hardware robusto, todo lo anterior a precios y tiempos de producción considerablemente bajos.

En la mayoría de las aplicaciones las redes neuronales suelen ser en su mayoría el reconocimiento de patrones, tal es el caso que expone Matich (2001): “buscar un patrón en una serie de ejemplos, clasificar patrones, completar una señal a partir de valores parciales o reconstruir el patrón correcto partiendo de uno distorsionado”. Aunque en el área de control las redes neuronales están tomando bastante terreno. Algunos tipos de aplicaciones de las redes neuronales son la asociación y clasificación, regeneración de patrones, regresión y generalización, así como optimización.

La asociación y clasificación consiste principalmente en que un clasificador sea capaz de reconocer y organizar versiones distorsionadas de un patrón dado, así como poder ser

inmune al ruido de la señal o patrón que entra al sistema para así recuperar un resultado más limpio.

La regeneración de patrones se trata esencialmente de que con un grupo de información vaga se pueda recuperar el patrón o la información faltante, la clave para que este tipo de reconocimiento es el uso adecuado de la información contextual, para resolver así dos tipos de problemáticas: los temporales o los estáticos.

Por otro lado, la regeneración y generalización, donde este último debe ser capaz de generar una salida correcta a partir de una entrada desconocida o que no se haya entrenado para dicha información anteriormente, esto mediante la detección de regularidades en las características de los datos introducidos, siendo esta parte bastante crítica y selectiva, creando así un sistema altamente eficaz en cualquier espacio a pesar de que el entrenamiento se haya realizado con elementos limitados.

Por último, la optimización consiste en que se busca el mínimo absoluto de una función de energía, en casos complejos se deben tomar en cuenta cientos de criterios, así como las limitantes que se tienen, pero existen ocasiones donde se pueden encontrar fácilmente.

## 2.6 Lenguajes de Programación

Conforme la computación se fue desarrollando, los lenguajes de programación también, ya que son las herramientas principales para el desarrollo de software, estos definen y administran los comportamientos tanto del hardware como de la lógica que posee la

computadora, su principal función es crear programas capaces de tener una comunicación entre la máquina y el usuario.

Un lenguaje de programación está conformado principalmente por una secuencia de símbolos y reglas de sintaxis la cual a su vez crea una estructura que se le da al lenguaje dándole un sentido a sus expresiones y elementos.

La computadora no entiende realmente los símbolos que los humanos toman como números o letras, por lo que se necesita una herramienta que traduzca dichos símbolos a lenguaje máquina (lenguaje que consta solo de 0 y 1) para que así se tenga un programa funcional y ejecutable.

Existe una clasificación de dichos lenguajes dependiendo de su aplicación y de cómo la computadora los traduce, hay tres grandes grupos de lenguajes los cuales son: el lenguaje máquina, el lenguaje de alto nivel y el lenguaje de bajo nivel.

El lenguaje máquina “Es el sistema de códigos interpretable directamente por un circuito microprogramable, como el microprocesador de una computadora” (Monteverde, s.f.), el lenguaje en si consiste en una serie de instrucciones que definen las acciones que realizará la máquina, las instrucciones se ejecutan por lo general en secuencia aunque estos pueden tener variantes por eventos externos o incluso por el mismo programa, este tipo de lenguaje es único en cada máquina o arquitectura aunque las instrucciones puedan ser muy parecidas entre ellas.

Por otro lado, el lenguaje de bajo nivel “es el que proporciona poca o ninguna abstracción del microprocesador de una computadora” (Monteverde, s.f.), por lo anterior hace que al momento de traducirse a lenguaje máquina esta sea muy sencilla, por lo general los programas de tipo ensamblador son los responsables de traducir un archivo fuente (escrito en lenguaje ensamblador) a un archivo que contenga un código máquina lo cual facilita su ejecución.

Por último, se tiene el lenguaje de alto nivel son instrucciones que se estructuran de manera similar a cómo se escribiera una instrucción a otro humano, es decir se genera un algoritmo más natural y sencillo de entender para los usuarios.

### 2.6.1 Python

Python es un lenguaje de alto nivel, creado principalmente para su facilidad a la hora de interpretarse, por lo general es utilizado con una programación orientada a objetos pero es capaz incluso de crear programación funcional o programación imperativa, además es un lenguaje multiplataforma esto significa que puede ser utilizado por un sinnúmero de sistemas operativos con total libertad, sin olvidar que además de lo anterior es un software libre esto quiere decir que no es necesario pagar para poder utilizarlo.

Algunas de las ventajas que tiene este lenguaje de programación son por ejemplo que es un lenguaje muy expresivo, ya que los códigos que se generan con él suelen ser cortos y entendibles en comparación a otros lenguajes como C o Java, incluso por esto muchos lo consideran lenguaje de alto nivel.

Así mismo, los programas suelen ser legibles, ayudando así a que los usuarios puedan comprender con facilidad lo que sucede en el programa y porqué sucede, además cuenta con un entorno interactivo perfecto para despejar dudas de errores o realizar pruebas. Si bien estas son algunas ventajas que hacen que muchos programadores se inclinen por el lenguaje Python, existen muchas otras más, así como muchas aplicaciones por el dinamismo con el que cuenta el lenguaje.

### 2.6.2 Lenguaje M

El lenguaje M se caracteriza principalmente por ser utilizado en el software de MATLAB el cual se centra en el estudio de sistemas discretos, continuos, no lineales y lineales.

Es un lenguaje sencillo, veloz y potente, esto pues no es necesario compilar o crear ejecutables, además los ficheros utilizados son de texto haciendo así que sean ligeros. Las funciones matemáticas y de aplicación están predefinidas y ordenadas en librerías (conocidas dentro del programa como *toolboxes*) estas funciones se pueden acceder fácilmente e incluso pueden ser modificadas por el mismo usuario, así como desarrollar sus propias funciones.

Es capaz de crear gráficos tanto en tres como en dos dimensiones, así como agregarles animaciones o efectos según sea necesario. Es posible intercambiar datos con otros lenguajes o entornos, así como acceder a diferentes dispositivos como tarjetas de sonido o adquisición de datos. Además, es posible desarrollar aplicaciones complejas las cuales pueden tener ventanas o menús según sea la aplicación y el caso que las necesite.

### **CAPÍTULO III: METODOLOGÍA**

En el presente capítulo se expondrá tanto el tipo como el diseño que se usa en la investigación, además se describirán de manera breve tanto el objeto de estudio como las variables a utilizar, así como los instrumentos y materiales empleados para el desarrollo físico, así mismo se explica concisamente el procedimiento que se llevará a cabo.

#### **3.1 Tipo de investigación.**

La actual investigación es de tipo descriptivo, esto pues este tipo de diseño describe cómo son y cómo se comportan ciertos fenómenos a estudiar. Texson, (2005) lo explica de la siguiente manera en su investigación: “este estudio mide o evalúa diferentes aspectos, tamaños o elementos del fenómeno a investigar, el principal interés de este estudio es

medir con la mayor precisión posible”. Dado que la presente investigación busca explicar la solución del problema inverso, es el más adecuado para exponer dicho procedimiento.

### 3.2 Diseño de la investigación.

El diseño de la actual investigación es de tipo no experimental, esto pues no se tiene un propósito específico del estudio más allá del análisis de lo que sucede en el fenómeno. García y Ruíz (2004), explican mejor lo que es este tipo de investigación “las variables no se manipulan intencionalmente, sino que sólo se observa y se analiza el fenómeno tal y como es en su contexto natural”, por ende, la investigación se centra en el uso de álgebra de números complejos, redes neuronales artificiales y software embebido para solucionar el problema inverso de un robot Delta plano, sin embargo, la investigación es además tipo no experimental, es de corte transversal, esto porque dichas investigaciones realizan sus pruebas en un tiempo único. Las autoras Hernández y Rojas, (2004) explican más concisamente lo que es una investigación de corte transversal en su investigación: “se recolectan datos en un solo momento, en un tiempo único, su propósito es describir variables y su incidencia de interrelación en un momento dado.”

### 3.3 Variables de estudio.

Variable de estudio	Definición conceptual de la variable
Rotación	Un sólido rígido realiza un movimiento de rotación cuando sus partículas describen circunferencias alrededor de una recta llamada eje de rotación. (Fernández, s. f.)
Distancia	Es la suma de las distancias recorridas. (Sepúlveda, 2016)
Desplazamiento	Es una magnitud vectorial que sólo depende de la posición inicial y final del cuerpo y es independiente de la trayectoria. (Fernández, s. f.)
Posición	El vector que une el lugar ocupado por el cuerpo con el origen del sistema de referencia. (Fernández, s. f.)
Movimiento	Es el cambio de posición de los cuerpos a lo largo del tiempo respecto a un sistema de referencia dado. (Fernández, s. f.)
Trayectoria	Es la línea geométrica que un cuerpo describe en su movimiento. (Fernández, s. f.)

Tabla 1. Variable de estudio (Espinoza, 2021)

### 3.4 Objetos de estudio.

El objeto de estudio de la presente investigación es la solución del problema inverso de un robot Delta (para el cual se utilizará el álgebra de números complejos, y redes neuronales artificiales con la finalidad de desarrollar sistemas mucho más eficaces a la hora de crear sistemas inteligentes con robots del tipo antes mencionado).

### 3.5 Materiales e instrumentos.

#### 3.5.1 MATERIALES

Para esta investigación se usaron el software Matlab y la plataforma de cálculo Excel.

#### 3.5.2 INSTRUMENTOS

El principal instrumento a utilizar para el presente trabajo fue el software Matlab, principalmente la toolbox designada para Inteligencia Artificial, así como Solidworks para crear el modelo, así como delimitar el espacio de trabajo en el cual se realizarían los experimentos.

### 3.6 Procedimiento

El procedimiento general de la presente investigación fue crear el modelado de mecanismos (Ver figura 12), en el cual se crea el modelado inverso del sistema, después se define una solución con el método de Newton-Raphson donde se desarrollan los cálculos del sistema inverso y directo, luego se desarrolla una validación de la solución mediante un dibujo en SolidWorks y una simulación en Matlab. Una vez hecho lo anterior se genera una base de datos de cinemática inversa para luego de normalizar el sistema, se definen los datos de entrada de la red, se diseña la topología de la red, al igual que se determina el algoritmo de

aprendizaje de la red neuronal, para posteriormente realizar el entrenamiento y por último realizar las pruebas pertinentes hasta conseguir el resultado deseado.

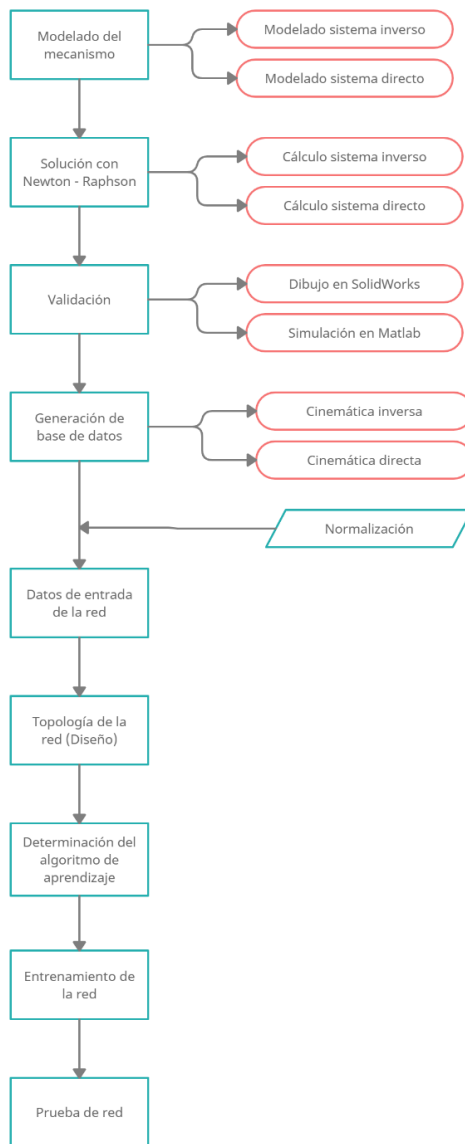


Figura 24. Diagrama de desarrollo. (Espinoza, 2021)

Los procedimientos específicos para realizar los estudios de redes neuronales se describen a continuación (Jiménez, Espinoza, Ruiz, Núñez y Reyes, 2022).

**a) Protocolo para obtener los datos de entrada para la red neuronal**

- 1) Identificar la configuración del área de trabajo del robot en Solidworks
- 2) Construir un mapa de puntos en toda o en una parte del área de trabajo del robot.  
Cada punto debe tener coordenadas (X,Y) medidas desde una referencia fija.
- 3) La totalidad de los puntos considerados debe ser lo suficientemente grande para entrenar la red sin llegar al punto de sobre entrenarla.
- 4) De preferencia el mapa de puntos puede ser una malla uniforme y simétrica, pero también puede ser no uniforme, no simétrica y menos ordenada.
- 5) Obtener cada coordenada (X, Y) del Solidworks directamente, o mediante una regla aritmética sacar todos los demás puntos a considerar.
- 6) Obtener en una tabla de Excel o equivalente donde se muestren las coordenadas de cada punto de preferencia en un orden específico.
- 7) Utilizar el programa de Matlab y el modelo cinemático inverso descrito en la configuración final para que lea esa lista de coordenadas y calcule, aplicando el método del Newton-Rapshon o equivalente, la cinemática inversa de cada punto considerado.
- 8) Hacer una tabla que tenga las coordenadas de los puntos de SolidWorks y sus 4 ángulos y 4 ejes calculados por Matlab.
- 9) Los datos de entrada a la red neuronal serán los datos del inciso 8).

**b) Protocolo para el entrenamiento de la red neuronal**

- 1) Diseñar una base de datos que tenga un vector de entradas y otro vector de salidas.
- 2) La base de datos debe estar integrada por todas aquellas configuraciones reales del robot. Se deben eliminar aquellas configuraciones espejo que sean generadas por el Newton-Raphson.
- 3) La base de datos debe tener una división: una base de datos para validar y una base de datos para entrenamiento.
- 4) Una vez depurada la base de datos, esta debe prepararse para el ingreso a la red, considerando normalizar la información tanto de entrada como de salida. Los rangos de las variables de entrada del robot delta están entre -1 y 1, por lo que no es necesario normalizar.
- 5) Se debe configurar la red, iniciando por definir su topología: (entradas, capas ocultas, número de neuronas por capa, capas de salida, salidas).
- 6) Se deben definir las funciones de activación (p.e. hyperbolic, tangent, sigmoid).
- 7) Para el caso de las redes de retroalimentación hacia adelante la función de rendimiento por default es el error cuadrático medio (mse).
- 8) Se debe seleccionar el algoritmo de entrenamiento (p.e Bayesian regularization algorithm o el Levenberg–Marquardt algorithm).
- 9) Se debe seleccionar el proceso de validación estadística de la red (p.e. Análisis de regresión).
- 10) Se debe evitar que la red se sobreentrene.

- 11) Una vez entrenada la red, esta debe validarse. Puede ser de dos maneras: a) Usando la base de datos que no se entrenó o 2) Diseñando una trayectoria sobre los puntos entrenados).
- 12) Se debe generar una base datos con la información de entrada y salida del Newton Rapshon y la generada por la red.
- 13) Se deben evaluar las diferencias entre los datos calculados por el Newton-Rapshon y los calculados por la red entrenada.
- 14) Se deben graficar los resultados.
- 15) Se deben escribir las conclusiones.

### **C) Experimentación**

- 1) Selección del objetivo a mejorar y optimizar
- 2) Formulación de la hipótesis
- 3) Selección de las variables a manipular
- 4) Descripción de las variables fijas
- 5) Definir los criterios de comparación
- 6) Definir el método de validación
- 7) Definir el número de corridas
- 8) Definir el criterio estadístico para la visualización de las corridas
- 9) Realizar el proceso de cómputo (entrenamiento)
- 10) Análisis de la información y comprobación de la hipótesis

11) Descripción del mejoramiento obtenido

12) Conclusiones

## **CAPÍTULO IV: RESULTADOS Y DISCUSIÓN.**

En el presente capítulo, se exponen los resultados obtenidos a lo largo del actual trabajo de tesis. En la primera sección se explica el modelado cinemático de posición y desplazamiento de un robot tipo delta planar tipo 5R desarrollado con la rotación usual de números complejos (Jiménez et al, 2021b).

### 4.1 Modelado del mecanismo.

#### 4.1.1 Modelado del mecanismo: configuración de referencia o inicial

A continuación, se explica el modelado del mecanismo 5R, también conocido como robot paralelo plano de 2GDL. La configuración del mecanismo a utilizar es la misma que se muestra en la Figura 25, donde se observa que se han definido tanto vectores de posición,

así como bases móviles. El modelado tiene como objetivo definir ecuaciones que permitan establecer las coordenadas del punto “pot”, tomando el origen “O” como referencia.

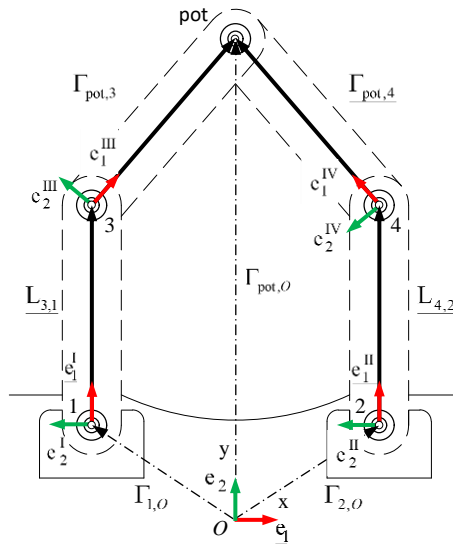


Figura 25. Configuración del mecanismo paralelo plano 5R y bases de la configuración inicial o no deformada (Jiménez *et al.* 2021b).

A partir de la Figura 25 se puede conocer que las coordenadas del punto “pot” se determinan por las siguientes ecuaciones, tomando como referencia el punto de origen “O”:

$$\underline{\Gamma_{pot,O}} = \underline{\Gamma_{1,O}} \oplus \underline{L_{3,1}} \oplus \underline{L_{pot,3}} \quad (1)$$

$$\underline{\Gamma_{pot,O}} = \underline{\Gamma_{2,O}} \oplus \underline{L_{4,2}} \oplus \underline{L_{pot,4}}$$

En su forma equivalente, sería:

$$\underline{\Gamma_{pot,O}} = \underline{\Gamma_{1,O}} \oplus \underline{L_{3,1}} \cdot \underline{e_1^I} \oplus \underline{L_{pot,3}} \cdot \underline{e_1^{III}} \quad (2)$$

$$\underline{\Gamma}_{pot,0} = \underline{\Gamma}_{2,0} \oplus L_{4,2} \cdot \underline{e}_1^I \oplus L_{pot,4} \cdot \underline{e}_1^{IV}$$

Ahora considere que las rotaciones de la base inercial  $\underline{e}_j = \{\underline{e}_1, \underline{e}_2\}$ , sobre las bases móviles se modelan utilizando la transformación lineal  $\rho: \mathfrak{R}^2 \rightarrow \mathfrak{R}^2$ . Dichas rotaciones se modelan como se muestra a continuación:

$$\underline{e}_1^I = \rho(p, \underline{e}_1) = \frac{1}{\|p\|} \cdot p \otimes e_1 \quad (3)$$

$$\underline{e}_1^{III} = \rho(r, \underline{e}_1) = \frac{1}{\|r\|} \cdot r \otimes e_1$$

$$\underline{e}_1^II = \rho(q, \underline{e}_1) = \frac{1}{\|q\|} \cdot q \otimes e_1$$

$$\underline{e}_1^{IV} = \rho(s, \underline{e}_1) = \frac{1}{\|s\|} \cdot s \otimes e_1$$

En este caso,  $p, q, r, s \in \mathfrak{R}^2$ , son números complejos de norma unitaria. Las expresiones (2) se escriben en términos de la base inercial fija y complejos de norma unitaria, esto es, en base a las expresiones (3):

$$\underline{\Gamma}_{pot,0} = \underline{\Gamma}_{1,0} \oplus L_{3,1} \cdot \{p \otimes \underline{e}_1\} \oplus L_{pot,3} \cdot \{r \otimes \underline{e}_1\} \quad (4)$$

$$\underline{\Gamma}_{pot,0} = \underline{\Gamma}_{2,0} \oplus L_{4,2} \cdot \{q \otimes \underline{e}_1\} \oplus L_{pot,4} \cdot \{s \otimes \underline{e}_1\}$$

Usando las operaciones  $\oplus: \mathfrak{R}^2 \times \mathfrak{R}^2 \rightarrow \mathfrak{R}^2$  y  $\otimes: \mathfrak{R}^2 \times \mathfrak{R}^2 \rightarrow \mathfrak{R}^2$  definidas en Jiménez et al. (2021), se consiguen las siguientes ecuaciones:

$$\Gamma_{pot,0,x} = \Gamma_{1,0,x} + L_{3,1,p_0} + L_{pot,3,r_0} \quad (5)$$

$$\Gamma_{pot,0,y} = \Gamma_{1,0,y} + L_{3,1,p_1} + L_{pot,3,r_1}$$

$$\Gamma_{pot,0,x} = \Gamma_{2,0,x} + L_{4,2,q_0} + L_{pot,4,s_0}$$

$$\Gamma_{pot,0,y} = \Gamma_{2,0,y} + L_{4,2,q_1} + L_{pot,4,s_1}$$

Posteriormente, se asocian las normas unitarias con los números complejos teniendo así:

$$p_0^2 + p_1^2 = 1; \quad r_0^2 + r_1^2 = 1 \quad (6)$$

$$q_0^2 + q_1^2 = 1; \quad s_0^2 + s_1^2 = 1$$

Luego se tienen las relaciones geométricas definidas entre los componentes de las rotaciones y los parámetros asociados con los números complejos  $p, q, r, s \in \mathfrak{R}^2$ , obteniendo:

$$p = (p_0, p_1); \quad p_0 = \cos\theta_1, \quad p_1 = \pm \text{sen}\theta_1 \quad (7)$$

$$q = (q_0, q_1); \quad q_0 = \cos\theta_2, \quad q_1 = \pm \text{sen}\theta_2$$

$$r = (r_0, r_1); \quad r_0 = \cos\theta_3, \quad r_1 = \pm \text{sen}\theta_3$$

$$s = (s_0, s_1); \quad s_0 = \cos\theta_4, \quad s_1 = \pm \text{sen}\theta_4$$

En este caso  $\theta_1, \theta_2, \theta_3, \theta_4 \in \mathfrak{R}$  se refieren a los desplazamientos angulares que se relacionan con las posiciones de los eslabones, así como con el eje cartesiano X, como se muestra en la Figura 19.

A continuación, se formula el problema directo relacionado con la configuración de referencia asociada con el mecanismo 5R motivo de estudio:

“Dado  $\Gamma_{pot,O}, \Gamma_{1,O}, \Gamma_{2,O} \in \mathfrak{R}^2, L_{3,1}, L_{4,2}, L_{pot,3}, L_{pot,4} \in \mathfrak{R}^+$  encuentre  $p = \{p_0, p_1\}, q = \{q_0, q_1\}, r = \{r_0, r_1\}, s = \{s_0, s_1\}$  tal que las expresiones (5) y (6) se satisfagan.”

El modelo que se relaciona con el problema cinemático inverso consiste en un sistema de 8 ecuaciones no lineales polinomiales y 8 incógnitas.

#### 4.1.2 Modelado de la configuración final.

En esta sección se modela lo que se denomina como la posición deformada o final del mecanismo 5R. La Figura 23 muestra la configuración final, así como sus bases móviles asociadas.

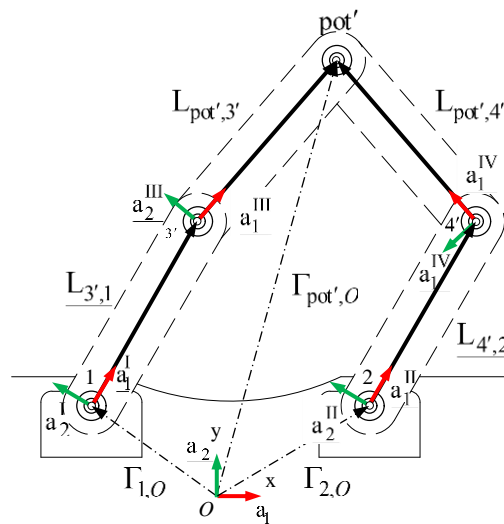


Figura 26. Bases móviles en la configuración final. (Jiménez et al, 2021b)

De acuerdo con la Figura 26, las coordenadas actualizadas del punto “pot” se obtienen a partir de las siguientes expresiones:

$$\underline{\Gamma}_{pot,0} = \underline{\Gamma}_{1,0} \oplus \underline{L}_{3,1} \oplus \underline{L}_{pot,3} \quad (8)$$

$$\underline{\Gamma}_{pot,0} = \underline{\Gamma}_{2,0} \oplus \underline{L}_{4,2} \oplus \underline{L}_{pot,4}$$

Las ecuaciones anteriores pueden ser escritas en términos de bases móviles obteniendo así:

$$\underline{\Gamma}_{pot,0} = \underline{\Gamma}_{1,0} \oplus \underline{L}_{3,1} \cdot \underline{a}_1^I \oplus \underline{L}_{pot,3} \cdot \underline{a}_1^{III} \quad (9)$$

$$\underline{\Gamma}_{pot,0} = \underline{\Gamma}_{2,0} \oplus \underline{L}_{4,2} \cdot \underline{a}_1^{II} \oplus \underline{L}_{pot,4} \cdot \underline{a}_1^{IV}$$

Las relaciones entre las bases móviles locales en la configuración de referencia (Ver figura 22) con respecto a las bases locales móviles descritas en la configuración deformada (ver figura 23), se pueden representar en términos de números complejos unitarios, teniendo:

$$\underline{a}_1^I = \rho(P, \underline{e}_1^I) = \rho(P, \rho(p, e_1)) = \frac{1}{\|P\|} \frac{1}{\|p\|} \cdot P \otimes p \otimes \underline{e}_1 = P \otimes p \otimes \underline{e}_1 \quad (10)$$

$$\underline{a}_1^{III} = \rho(R, \underline{e}_1^{III}) = \rho(R, \rho(r, e_1)) = \frac{1}{\|R\|} \frac{1}{\|r\|} \cdot R \otimes r \otimes \underline{e}_1 = R \otimes r \otimes \underline{e}_1$$

$$\underline{a}_1^{II} = \rho(Q, \underline{e}_1^{II}) = \rho(Q, \rho(q, e_1)) = \frac{1}{\|Q\|} \frac{1}{\|q\|} \cdot Q \otimes q \otimes \underline{e}_1 = Q \otimes p \otimes \underline{e}_1$$

$$\underline{a}_1^{IV} = \rho(S, \underline{e}_1^{IV}) = \rho(S, \rho(s, e_1)) = \frac{1}{\|S\|} \frac{1}{\|s\|} \cdot S \otimes s \otimes \underline{e}_1 = S \otimes s \otimes \underline{e}_1$$

En consecuencia, las ecuaciones de lazo (8) se escriben en términos de los complejos unitarios, consiguiendo:

$$\underline{\Gamma}_{pot',0} = \underline{\Gamma}_{1,0} \oplus \underline{L}_{3',1} \cdot \{P \otimes p \otimes \underline{e}_1\} \oplus \underline{L}_{pot',3'} \cdot \{R \otimes r \otimes \underline{e}_1\} \quad (11)$$

$$\underline{\Gamma_{pot',O}} = \underline{\Gamma_{2,O}} \oplus L_{4',2} \cdot \{Q \otimes q \otimes \underline{e_1}\} \oplus L_{pot',4'} \cdot \{S \otimes s \otimes \underline{e_1}\}$$

En este caso,  $P, Q, R, S \in \mathfrak{R}^2$  son los números complejos relacionados con la configuración final o deformada, la forma explícita de las expresiones (11) se muestra a continuación:

$$\Gamma_{pot',O,x} = \Gamma_{1,O,x} + L_{3',1}(P_0 p_0 - P_1 p_1) + L_{pot',3'}(R_0 r_0 - R_1 r_1) \quad (12)$$

$$\Gamma_{pot',O,y} = \Gamma_{1,O,y} + L_{3',1}(P_0 p_1 + P_1 p_0) + L_{pot',3'}(R_0 r_1 + R_1 r_0)$$

$$\Gamma_{pot',O,x} = \Gamma_{2,O,x} + L_{4',2}(Q_0 q_0 - Q_1 q_1) + L_{pot',4'}(S_0 s_0 - S_1 s_1)$$

$$\Gamma_{pot',O,y} = \Gamma_{2,O,y} + L_{4',2}(Q_0 q_1 + Q_1 q_0) + L_{pot',4'}(S_0 s_1 + S_1 s_0)$$

Las expresiones de forma unitaria son:

$$||P|| = P_0^2 + P_1^2 = 1; \quad ||Q|| = Q_0^2 + Q_1^2 = 1 \quad (13)$$

$$||R|| = R_0^2 + R_1^2 = 1; \quad ||S|| = S_0^2 + S_1^2 = 1$$

Por su parte, las relaciones geométricas definidas entre las componentes de las rotaciones y los parámetros asociados con los números complejos de la configuración deformada son las siguientes:

$$P = (P_0, P_1); \quad P_0 = \cos\alpha_1; \quad P_1 = \pm \text{sen}\alpha_1 \quad (14)$$

$$Q = (Q_0, Q_1); \quad Q_0 = \cos\alpha_2; \quad Q_1 = \pm \text{sen}\alpha_2$$

$$R = (R_0, R_1); \quad R_0 = \cos\alpha_3; \quad R_1 = \pm \text{sen}\alpha_3$$

$$S = (S_0, S_1); \quad S_0 = \cos\alpha_4; \quad S_1 = \pm \text{sen}\alpha_4$$

Aquí  $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathfrak{R}$  son los desplazamientos angulares definidos entre la configuración inicial y la configuración final, como se muestra en la Figura 27.

El problema cinemático inverso se enuncia de la siguiente forma:

“Dado  $\underline{\Gamma}_{pot,0}, \underline{\Gamma}_{1,0}, \underline{\Gamma}_{2,0} \in \mathfrak{R}^2, L_{3,1}, L_{4,2}, L_{pot,3}, L_{pot,4} \in \mathfrak{R}^+, p = \{p_0, p_1\}, q = \{q_0, q_1\}, r = \{r_0, r_1\}, s = \{s_0, s_1\}$  con  $\|p\| = \|q\| = \|r\| = \|s\|$ , encuentre:  $P = \{P_0, P_1\}, Q = \{Q_0, Q_1\}, R = \{R_0, R_1\}, S = \{S_0, S_1\}$  tal que las expresiones (12) y (13) se satisfagan.”

El problema de la cinemática inversa asociado con el mecanismo 5R motivo de estudio genera un modelo de 8 ecuaciones no lineales polinomiales y 8 incógnitas.

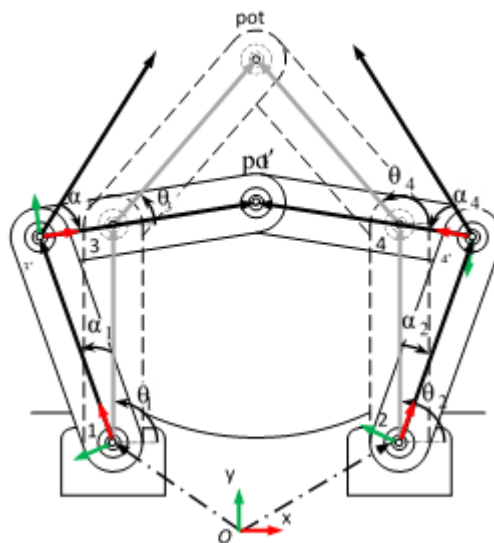


Figura 27. Desplazamientos angulares en la configuración inicial y final (Jiménez et al, 2021b)

### 4.1.3 Modelado de desplazamiento.

Para determinar el modelo de desplazamiento es necesario poner en función del tiempo las siguientes funciones.

$$\underline{\Gamma}_{pot',0} = \underline{\Gamma}_{1,0} \oplus \underline{L}_{3',1} \oplus \underline{L}_{pot',3'} \quad (15)$$

$$\underline{\Gamma}_{pot',0} = \underline{\Gamma}_{2,0} \oplus \underline{L}_{4',2} \oplus \underline{L}_{pot',4'}$$

O, en términos de las bases móviles, teniendo así:

$$\underline{\Gamma}_{pot',0} = \underline{\Gamma}_{1,0} \oplus \underline{L}_{3',1} \cdot \underline{a}_1^I \oplus \underline{L}_{pot',3'} \cdot \underline{a}_1^{III} \quad (16)$$

$$\underline{\Gamma}_{pot',0} = \underline{\Gamma}_{2,0} \oplus \underline{L}_{4',2} \cdot \underline{a}_1^{II} \oplus \underline{L}_{pot',4'} \cdot \underline{a}_1^{IV}$$

Las bases móviles se definen de la siguiente manera:

$$\underline{a}_1^I(t) = \rho(P(t), \underline{e}_1) = \rho(P(t), \rho(p, e_1)) = \frac{1}{\|P(t)\|} \frac{1}{\|p\|} \cdot P(t) \otimes p \otimes \underline{e}_1 = P(t) \otimes p \otimes \underline{e}_1 \quad (17)$$

$$\underline{a}_1^{III}(t) = \rho(R(t), \underline{e}_1^{III}) = \rho(R(t), \rho(r, e_1)) = \frac{1}{\|R(t)\|} \frac{1}{\|r\|} \cdot R(t) \otimes r \otimes \underline{e}_1 = R(t) \otimes r \otimes \underline{e}_1$$

$$\underline{a}_1^{II}(t) = \rho(Q(t), \underline{e}_1^{II}) = \rho(Q(t), \rho(q, e_1)) = \frac{1}{\|Q(t)\|} \frac{1}{\|q\|} \cdot Q(t) \otimes q \otimes \underline{e}_1 = Q(t) \otimes p \otimes \underline{e}_1$$

$$\underline{a}_1^{IV}(t) = \rho(S(t), \underline{e}_1^{IV}) = \rho(S(t), \rho(s, e_1)) = \frac{1}{\|S(t)\|} \frac{1}{\|s\|} \cdot S(t) \otimes s \otimes \underline{e}_1 = S(t) \otimes s \otimes \underline{e}_1$$

De modo que, las ecuaciones de lazo (15) se escriben en términos de los complejos unitarios, esto es:

$$\underline{\Gamma}_{pot',O}(t) = \underline{\Gamma}_{1,O} \oplus L_{3',1} \cdot \{P(t) \otimes p \otimes \underline{e}_1\} \oplus L_{pot',3'} \cdot \{R(t) \otimes r \otimes \underline{e}_1\} \quad (18)$$

$$\underline{\Gamma}_{pot',O}(t) = \underline{\Gamma}_{2,O} \oplus L_{4',2} \cdot \{Q(t) \otimes q \otimes \underline{e}_1\} \oplus L_{pot',4'} \cdot \{S(t) \otimes s \otimes \underline{e}_1\}$$

En este caso,  $P(t), Q(t), R(t), S(t) \in \mathfrak{R}^2$  son los números complejos relacionados con la configuración final en términos del tiempo. La forma explícita de las expresiones (18) es la siguiente:

$$\Gamma_{pot',O,x}(t) = \Gamma_{1,O,x} + L_{3',1}(P_0(t)p_0 - P_1(t)p_1) + L_{pot',3'}(R_0(t)r_0 - R_1(t)r_1) \quad (19)$$

$$\Gamma_{pot',O,y}(t) = \Gamma_{1,O,y} + L_{3',1}(P_0(t)p_1 + P_1(t)p_0) + L_{pot',3'}(R_0(t)r_1 + R_1(t)r_0)$$

$$\Gamma_{pot',O,x}(t) = \Gamma_{2,O,x} + L_{4',2}(Q_0(t)q_0 - Q_1(t)q_1) + L_{pot',4'}(S_0(t)s_0 - S_1(t)s_1)$$

$$\Gamma_{pot',O,y}(t) = \Gamma_{2,O,y} + L_{4',2}(Q_0(t)q_1 + Q_1(t)q_0) + L_{pot',4'}(S_0(t)s_1 + S_1(t)s_0)$$

Así como:

$$||P(t)|| = P_0^2(t) + P_1^2(t) = 1; \quad ||Q(t)|| = Q_0^2(t) + Q_1^2(t) = 1 \quad (20)$$

$$||R(t)|| = R_0^2(t) + R_1^2(t) = 1; \quad ||S(t)|| = S_0^2(t) + S_1^2(t) = 1$$

En consecuencia, las relaciones geométricas definidas entre las componentes de las rotaciones y los parámetros asociados con los números complejos de la configuración deformada en función del tiempo son:

$$P(t) = (P_0(t), P_1(t)); \quad P_0(t) = \cos\alpha_1(t); \quad P_1(t) = \pm \text{sen}\alpha_1(t) \quad (21)$$

$$Q(t) = (Q_0(t), Q_1(t)); \quad Q_0(t) = \cos\alpha_2(t); \quad Q_1(t) = \pm \text{sen}\alpha_2(t)$$

$$R(t) = (R_0(t), R_1(t)); \quad R_0(t) = \cos\alpha_3(t); \quad R_1(t) = \pm \text{sen}\alpha_3(t)$$

$$S(t) = (S_0(t), S_1(t)); \quad S_0(t) = \cos\alpha_4(t); \quad S_1(t) = \pm \text{sen}\alpha_4(t)$$

Aquí  $\alpha_1(t), \alpha_2(t), \alpha_3(t), \alpha_4(t) \in \mathfrak{R}$  son los desplazamientos angulares definidos entre la configuración inicial y la configuración final.

El problema cinemático inverso se expresa de la siguiente forma:

“Dado  $\Gamma_{pot,0}(t) \in \mathfrak{R}, \Gamma_{1,0}, \Gamma_{2,0} \in \mathfrak{R}^2, L_{3,1}, L_{4,2}, L_{pot,3}, L_{pot,4} \in \mathfrak{R}^+, p = \{p_0, p_1\}, q = \{q_0, q_1\}, r = \{r_0, r_1\}, s = \{s_0, s_1\}$  con  $\|p\| = \|q\| = \|r\| = \|s\|$ , encuentre:  $P(t) = \{P_0(t), P_1(t)\}, Q(t) = \{Q_0(t), Q_1(t)\}, R(t) = \{R_0(t), R_1(t)\}, S(t) = \{S_0(t), S_1(t)\}$  tal que las expresiones (19) y (20) se satisfagan.”

El problema de la cinemática inversa asociado con el mecanismo 5R motivo de estudio genera un modelo de 8 ecuaciones no lineales polinomiales y 8 incógnitas.

#### 4.2 Entrenamiento de una red neuronal artificial.

En esta sección se presenta el proceso de entrenamiento de una red neuronal especializada en resolver el problema cinemático inverso relacionado con el robot motivo de estudio en esta tesis. Ciertamente el problema inverso asociado con el robot delta fue estudiado por Jiménez *et al.* (2021b), sin embargo, en dicho trabajo no se presentó una experimentación

computacional relacionada con determinar la mejor red sobre un conjunto de propuestas. En este trabajo se desarrollará una experimentación para determinar bajo criterios establecidos una arquitectura ideal de redes neuronales que se distinga por tener las mejores características dentro de un conjunto de propuestas. El desarrollo de la experimentación se llevará a cabo siguiendo las fases siguientes:

a) FASE 1. CONSTRUCCIÓN DE UN MODELO REPRESENTATIVO

Esta fase se refiere a la construcción de un modelo de entrenamiento representativo, el cual consiste en diseñar una base de datos o el target de entradas y salidas a entrenar, la normalización de los datos, la construcción y la topología de la red, la selección de un algoritmo de entrenamiento, la parametrización, las corridas, la generación de resultados y las verificaciones de los resultados según una referencia de representatividad. Para generar un modelo representativo es necesario aplicar los pasos descritos por Jiménez et al. (2021a). En esta fase se aplicarán los dos primeros procedimientos específicos descritos al final del capítulo 3 (Jiménez, Espinoza, Ruiz, Pérez y Reyes, 2022).

b) FASE 2. DISEÑO DE LAS PRUEBAS EXPERIMENTALES

En esta fase se diseña las pruebas experimentales necesarias para alcanzar un cierto objetivo tomando como referencia el modelo representativo de la fase anterior, y tiene que ver con la selección y la definición de variables que pueden manipularse para determinar los cambios en otras ya sea internas o externas. Esta fase utiliza el último procedimiento descrito al final del capítulo 3.

#### 4.2.1 Aplicación de los protocolos.

En esta sección se aplicarán los protocolos descritos en el capítulo 3 de este trabajo de tesis. El primer protocolo en ser utilizado corresponde a la preparación de los datos de entrada para entrenar la red. Esta fase es de suma importancia debido a que de la calidad y la cantidad de datos que se recolecten depende la calidad del entrenamiento de la red neuronal. Posteriormente, con la información recabada, se aplica otro protocolo cuya función es la de entregar un modelo representativo de un entrenamiento. Este modelo será la base para que, pueda ser aplicado el protocolo relacionado con la experimentación el cual consiste en obtener una topología de una red mejorada.

##### 4.2.1.1 Protocolo para obtener los datos de entrada para la red neuronal

La obtención de la base de datos es la prioridad del primer protocolo. Para el caso del robot en estudio, la base de datos se generará usando el modelo generado por las ecuaciones (19) y (20):

$$\Gamma_{pot',0,x} = \Gamma_{1,0,x} + L_{3',1}(P_0(t)p_0 - P_1(t)p_1) + L_{pot',3'}(R_0(t)r_0 - R_1(t)r_1) \quad (19)$$

$$\Gamma_{pot',0,y} = \Gamma_{1,0,y} + L_{3',1}(P_0(t)p_1 + P_1(t)p_0) + L_{pot',3'}(R_0(t)r_1 + R_1(t)r_0)$$

$$\Gamma_{pot',0,x} = \Gamma_{2,0,x} + L_{4',2}(Q_0(t)q_0 - Q_1(t)q_1) + L_{pot',4'}(S_0(t)s_0 - S_1(t)s_1)$$

$$\Gamma_{pot',0,y} = \Gamma_{2,0,y} + L_{4',2}(Q_0(t)q_1 + Q_1(t)q_0) + L_{pot',4'}(S_0(t)s_1 + S_1(t)s_0)$$

$$\|P(t)\| = P_0^2(t) + P_1^2(t) = 1; \quad \|Q(t)\| = Q_0^2(t) + Q_1^2(t) = 1 \quad (20)$$

$$\|R(t)\| = R_0^2(t) + R_1^2(t) = 1; \quad \|S(t)\| = S_0^2(t) + S_1^2(t) = 1$$

Este modelo es no lineal y se caracteriza por formar un sistema de 8 ecuaciones no lineales con 8 incógnitas, para lo cual es necesario utilizar un método numérico, que en este caso se usó el método de Newton-Rapshon.

La aplicación del protocolo se llevará paso a paso. Esto es:

*1) Identificar la configuración del área de trabajo del robot en Solidworks*

Para lograr obtener los datos de las coordenadas que pueden ser utilizadas para generar la base de datos, es necesario considerar toda el área de trabajo del robot o en su caso una porción de ella. La Figura 28 muestra la zona del área de trabajo, en este caso delimitada por un rectángulo, en donde se recolectarán las coordenadas de los puntos para entrenar la red.

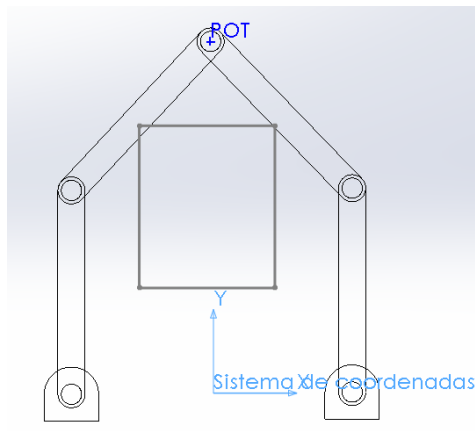


Figura 28. Localización de la zona de recolección de las coordenadas de los puntos en el área de trabajo.

2) Construir un mapa de puntos en toda o en una parte del área de trabajo del robot.

Cada punto debe tener coordenadas (X,Y) medidas desde una referencia fija.

Sobre la porción del área de trabajo seleccionada se debe localizar un mapa de puntos por los cuales el órgano terminal del robot (pot) se posará y para lo cual se deberá calcular el problema inverso para cada punto. La Figura 29 muestra una porción de los puntos que serán considerados para el entrenamiento.

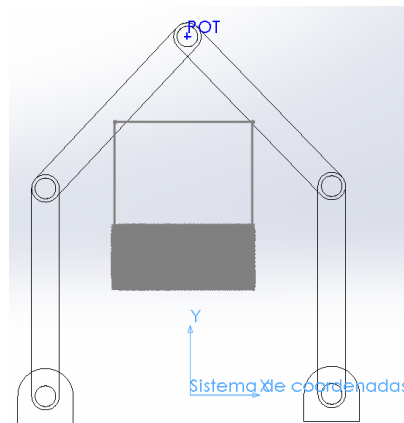


Figura 29. Zona de localización de los puntos.

3) La totalidad de los puntos considerados debe ser lo suficientemente grande para poder entrenar los datos.

La cantidad de datos a considerar es un factor importante para el buen entrenamiento de la red. Para el caso del robot motivo de estudio se consideraron un total de 10,000 puntos sobre la porción del área de trabajo seleccionada y sus respectivas coordenadas. Es decir, sobre 10,000 puntos se deberá resolver el problema inverso para obtener los datos.

- 4) De preferencia el mapa de puntos puede ser una malla uniforme y simétrica, pero también puede ser no uniforme, no simétrica y menos ordenada.

El mapa de puntos considerado es una malla uniforme y se muestra en la Figura 30.

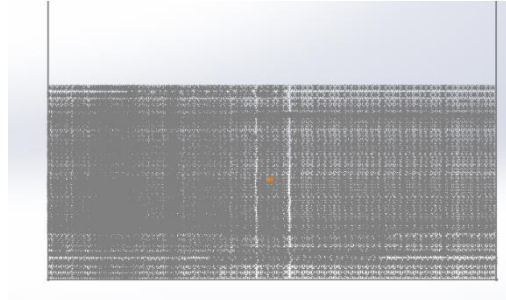


Figura 30. Malla de puntos.

- 5) Obtener cada coordenada (X, Y) del Solidworks directamente, o mediante una regla aritmética sacar todos los demás puntos a considerar.

Una vez obtenida el área de trabajo, se tomaron puntos alrededor del perímetro de este para posteriormente sumar 0.001, este valor se decidió para así seguir un patrón específico en el eje Y, siempre respetando las coordenadas del perímetro esto para evitar datos que para esta prueba no son necesarios.

- 6) Obtener en una tabla de Excel o equivalente donde se muestren las coordenadas de cada punto de preferencia en un orden específico.

La Tabla 2 muestra una lista de las coordenadas solo de 10 puntos.

rpotOx	rpotOy
43.04	78.801
41.28	78.802
39.87	78.803
37.94	78.804
35.68	78.805
32.97	78.806
30.93	78.807
28.59	78.808
27.1	78.809

Tabla 2 Tabla de coordenadas de los puntos sobre la zona de entrenamiento del robot

- 6) Utilizar el programa de Matlab y el modelo cinemático inverso descrito en la configuración final para que lea esa lista de coordenadas y calcule, aplicando el método del Newton-Rapshon o equivalente, la cinemática inversa de cada punto considerado. (Ver anexo 1)

El programa toma los datos de entrada (puntos tomados del modelo en Solidworks) los cuales entran a un bucle en el cual se definen parámetros para que este trabaje de la forma más óptima y genere los resultados de la cinemática inversa, en la cual se definen tanto los vectores con las funciones necesarias como la matriz jacobiana, una vez obteniendo lo anterior, se obtienen los resultados tanto de los ángulos como de las posiciones del robot.

La Tabla 3 muestra un cálculo de dos puntos usando el modelo del problema inverso en la configuración final. El cálculo genera los ángulos de rotación y los parámetros de cada complejo.

rpotOx	rpotOy	phi1	phi2	phi3	phi4	P0	P1	Q0	Q1	R0	R1	S0	S1
43.04	78.801	86.61243	52.3271863	-27.1721172	-165.526868	0.05908981	0.99825267	0.88963871	-0.45666504	0.61115154	0.79151361	-0.96826495	-0.24992598
41.28	78.802	87.2881854	53.4139336	-27.2120929	-164.848753	0.04731242	0.99888014	0.88931988	-0.45728564	0.59602962	0.80296245	-0.96523924	-0.26136794

Tabla 3. Cálculo para dos puntos de la cinemática inversa.

8) Hacer una tabla que tenga las coordenadas de los puntos de SolidWorks y sus 4 ángulos y 4 ejes calculados por Matlab.

La Tabla 4 muestra una lista de la base de datos generadas (solo 10). La función en donde se programa el modelo matemático en Matlab es (FiveAR\_mechanism\_inverse\_kinematics)

rpotOx	rpotOy	phi1	phi2	phi3	phi4	P0	P1	Q0	Q1	R0	R1	S0	S1
43.04	78.801	86.61243	52.3271863	-27.1721172	-165.526868	0.05908981	0.99825267	0.88963871	-0.45666504	0.61115154	0.79151361	-0.96826495	-0.24992598
41.28	78.802	87.2881854	53.4139336	-27.2120929	-164.848753	0.04731242	0.99888014	0.88931988	-0.45728564	0.59602962	0.80296245	-0.96523924	-0.26136794
39.87	78.803	87.8322059	54.2749405	-27.237694	-164.322234	0.03782612	0.99928434	0.88911546	-0.45768296	0.58389634	0.81182823	-0.96279668	-0.27022685
37.94	78.804	88.5807207	55.439761	-27.2636026	-163.625052	0.02476856	0.99969321	0.88890841	-0.45808496	0.56727238	0.82353023	-0.95943733	-0.28192198
35.68	78.805	89.4632505	56.7836614	-27.280107	-162.842881	0.00936791	0.99995612	0.88877642	-0.458341	0.54780181	0.83660814	-0.95549941	-0.29499301
32.97	78.806	90.530598	58.3668852	-27.2797284	-161.952704	-0.00926055	0.99995712	0.88877945	-0.45833513	0.52447809	0.85142395	-0.95080111	-0.30980195
30.93	78.807	91.341062	59.5385554	-27.2643931	-161.316413	-0.02340381	0.99972609	0.88890209	-0.45809723	0.50695844	0.8619705	-0.94730208	-0.32034163
28.59	78.808	92.2784457	60.8616717	-27.2306522	-160.62127	-0.0397559	0.99920942	0.88917171	-0.45757368	0.48691979	0.87344669	-0.9433459	-0.33181095
27.1	78.809	92.8799198	61.692651	-27.1997738	-160.197807	-0.05024292	0.99873703	0.88941818	-0.45709442	0.47420114	0.88041654	-0.94086781	-0.33877393

Tabla 4. Cálculo para 10 puntos de 10000 puntos de la cinemática inversa.

10) Los datos de entrada a la red neuronal serán los datos del inciso 8).

#### 4.2.1.2 Entrenamiento para el modelo representativo.

En esta sección se presentan el entrenamiento de la red neuronal para la base de datos generada en el protocolo anterior. El proceso se llevará a cabo siguiendo los pasos siguientes:

##### a) *Protocolo para el entrenamiento de la red neuronal*

1) Diseñar una base de datos que tenga un vector de entradas y otro vector de salidas.

Para el robot, el vector de entradas son las coordenadas de una trayectoria o un conjunto de puntos dentro del área de trabajo que puedan ser alcanzables por el órgano terminal (ver Tabla 5). Para el caso del vector de salidas este debe estar compuesto por los parámetros que se calculan a partir de cada coordenada del

punto del vector de entradas usando el Newton-Rapshon (para el caso del robot delta estos son los 4 parámetros de rotación y 4 ejes).

rpotOx	rpotOy	phi1	phi2	phi3	phi4	P0	P1	Q0	Q1	R0	R1	S0	S1
43.04	78.801	86.61243	52.3271863	-27.1721172	-165.526868	0.05908981	0.99825267	0.88963871	-0.45666504	0.61115154	0.79151361	-0.96826495	-0.24992598
41.28	78.802	87.2881854	53.4139336	-27.2120929	-164.848753	0.04731242	0.99888014	0.88931988	-0.45728564	0.59602962	0.80296245	-0.96523924	-0.26136794
39.87	78.803	87.8322059	54.2749405	-27.237694	-164.322234	0.03782612	0.99928434	0.88911546	-0.45768296	0.58389634	0.81182823	-0.96279668	-0.27022685
37.94	78.804	88.5807207	55.439761	-27.2636026	-163.625052	0.02476856	0.99969321	0.88890841	-0.45808496	0.56727238	0.82353023	-0.95943733	-0.28192198
35.68	78.805	89.4632505	56.7836614	-27.280107	-162.842881	0.00936791	0.99995612	0.88877642	-0.458341	0.54780181	0.83660814	-0.95549941	-0.29499301
32.97	78.806	90.530598	58.3668852	-27.2797284	-161.952704	-0.00926055	0.99995712	0.88877945	-0.45833513	0.52447809	0.85142395	-0.95080111	-0.30980195
30.93	78.807	91.341062	59.5385554	-27.2643931	-161.316413	-0.02340381	0.99972609	0.88890209	-0.45809723	0.50695844	0.8619705	-0.94730208	-0.32034163
28.59	78.808	92.2784457	60.8616717	-27.2306522	-160.62127	-0.0397559	0.99920942	0.88917171	-0.45757368	0.48691979	0.87344669	-0.9433459	-0.33181095
27.1	78.809	92.8799198	61.692651	-27.1997738	-160.197807	-0.05024292	0.99873703	0.88941818	-0.45709442	0.47420114	0.88041654	-0.94086781	-0.33877393
24.16	78.81	94.0773522	63.3069044	-27.1176735	-159.404016	-0.07110317	0.99746897	0.89007224	-0.45581948	0.44921134	0.89342553	-0.93608419	-0.35177604
22.64	78.811	94.7024175	64.1283243	-27.0634911	-159.015385	-0.08198056	0.99663393	0.8905029	-0.45497757	0.43635704	0.8997736	-0.93367662	-0.35811725
20.7	78.812	95.506231	65.1641288	-26.9827702	-158.540007	-0.095954	0.99538577	0.89114301	-0.45372254	0.42002033	0.90751469	-0.93067325	-0.36585146
19.45	78.813	96.027964	65.824075	-26.9235137	-158.246006	-0.10501384	0.99447076	0.89161178	-0.45280066	0.40953974	0.91229228	-0.92878372	-0.37062218
17.82	78.814	96.7127829	66.6761838	-26.8378839	-157.876459	-0.11689231	0.9931446	0.8922875	-0.45146762	0.39592724	0.91828189	-0.92637398	-0.37660491
16.68	78.815	97.1949388	67.2664323	-26.7720406	-157.627413	-0.1252456	0.99212577	0.89280573	-0.45044192	0.38644646	0.92231184	-0.92472825	-0.38062799
14.87	78.816	97.9658091	68.1943499	-26.6577576	-157.247048	-0.13858214	0.99035094	0.89370241	-0.44866022	0.3714594	0.9284492	-0.92218104	-0.38675848
13.16	78.817	98.7004521	69.0606545	-26.538246	-156.904699	-0.15126862	0.98849269	0.89463632	-0.4467951	0.35737944	0.93395928	-0.91985367	-0.39226168
11.6	78.818	99.3762288	69.8423874	-26.4192039	-156.606501	-0.16291663	0.98663984	0.89556268	-0.44493537	0.34460381	0.93874822	-0.91779968	-0.39704375

Tabla 5. Cálculo para 20 puntos de 10000 puntos de la cinemática inversa.

- 2) La base de datos debe estar integrada por todas aquellas configuraciones reales del robot. Se deben eliminar aquellas configuraciones espejo que sean generadas por el Newton-Rapshon.

La figura 31 muestra las configuraciones espejo que se generan debido a que el sistema de ecuaciones es no lineal y tiene dos soluciones para el mismo punto. Estas configuraciones fueron eliminadas de la base de datos.

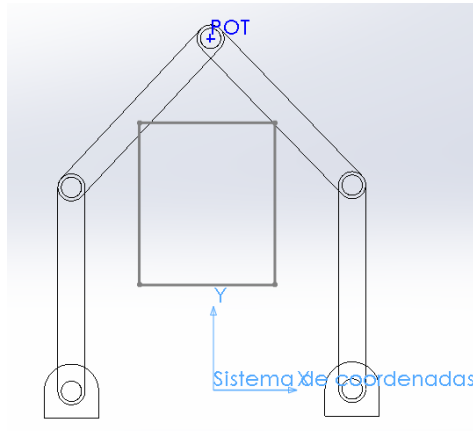


Figura 31 Configuraciones espejo del robot

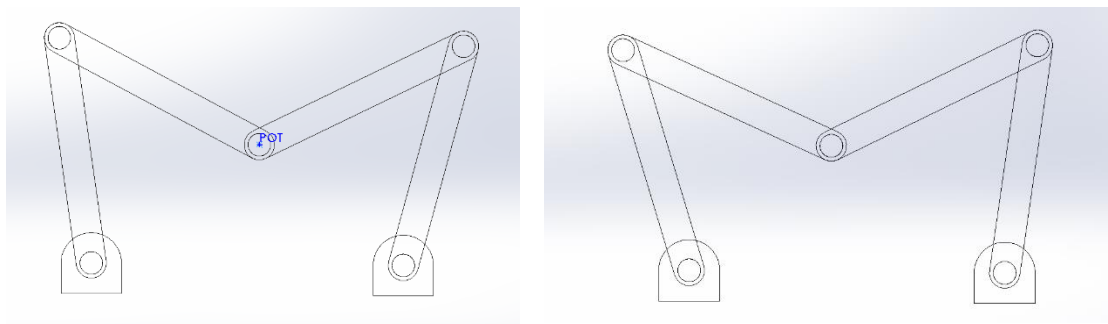


Figura 32. Configuraciones para alcanzar el punto (89.9, 76.6)

- 3) La base de datos debe tener una división: una base de datos para validar y una base de datos para entrenamiento.

Se creó una base de datos de entrenamiento la cual se generó a partir del código del anexo 2

Este código genera los resultados a partir de iteraciones de cálculos arrojando así los datos necesarios.

- 4) Una vez depurada la base de datos, esta debe prepararse para el ingreso a la red, considerando normalizar la información tanto de entrada como de salida. Los rangos de las variables de entrada del robot delta están entre -1 y 1, por lo que no es necesario normalizar.

$$-1 \leq p_0 \leq 1$$

$$-1 \leq p_1 \leq 1$$

$$-1 \leq q_0 \leq 1$$

$$-1 \leq q_1 \leq 1$$

$$-1 \leq r_0 \leq 1$$

$$-1 \leq r_1 \leq 1$$

$$-1 \leq s_0 \leq 1$$

$$-1 \leq s_1 \leq 1$$

$$-1 \leq ph_1 \leq 1$$

$$-1 \leq ph_2 \leq 1$$

$$-1 \leq ph_3 \leq 1$$

$$-1 \leq ph_4 \leq 1$$

- 5) Se debe configurar la red, iniciando por definir su topología: (entradas, capas ocultas, número de neuronas por capa, capas de salida, salidas).

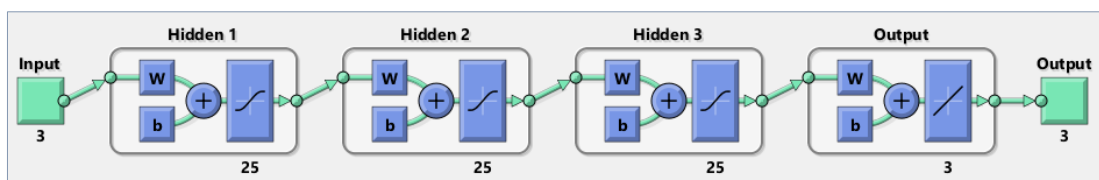


Figura 33. Topología de la red neuronal para el modelo Representativo.

	A	B	C	D
1	Input layer	Neuron	Output layer	Output
2		2	3	3
3		2	3	3
4		2	3	3
5		2	4	3
6		2	4	3
7		2	4	3
8		2	5	3
9		2	5	3
10		2	5	3
11		2	3	3
12		2	3	3
13		2	3	3
14		2	4	3
15		2	4	3
16		2	4	3
17		2	5	3
18		2	5	3
19		2	5	3
20		2	3	3

Tabla 6. Tabla que representa las combinaciones de topologías que se usaron en el modelo representativo, para 20 puntos.

- 6) Se deben definir las funciones de activación (p.e. hyperbolic tangent sigmoid).

Se decidió utilizar la función de activación Hyperbolic tangent sigmoid. Pues tiene un rango de salida de -1 a 1 teniendo un aumento monótono y gradual, facilitando la optimización, también la derivación de esta función hace que sea prácticamente una derivación directa siendo así mucho más eficiente.

- 7) Para el caso de las redes de retroalimentación hacia adelante la función de rendimiento por default es el error cuadrático medio (mse).

$$Error = \frac{Modelo Matemático - Red Neuronal}{100}$$

- 8) Se debe seleccionar el algoritmo de entrenamiento (p.e Bayesian regularization algorithm o el Levenberg–Marquardt algorithm).

Se seleccionó el algoritmo de entrenamiento Bayesian Regulation, como se ve en la figura 34.

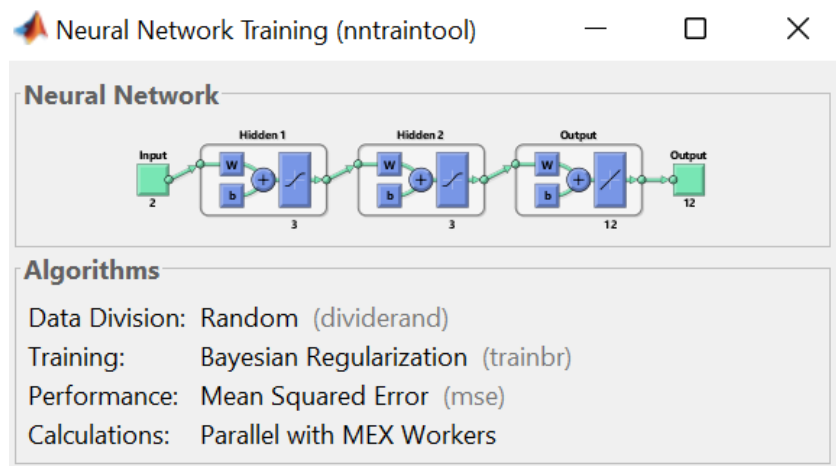


Figura 34. Topología y algoritmo utilizado para entrenamiento de la red.

9) Se debe procesar la red (Corrida de datos).

La figura 35 muestra la pantalla de Matlab con los resultados de la corrida de datos.

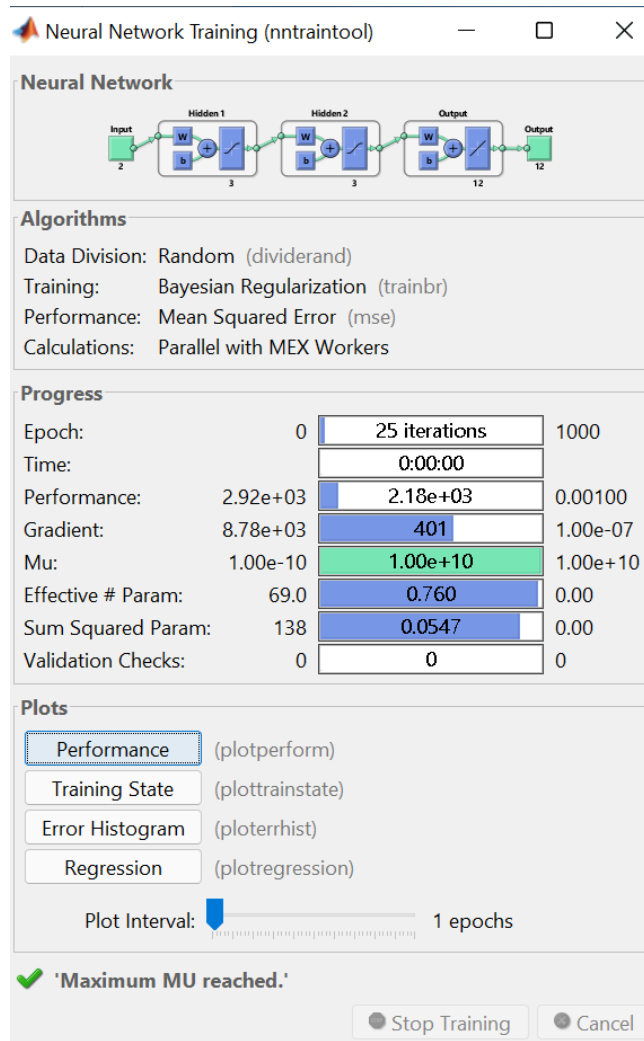


Figura 35. Pantalla de Matlab con los resultados de la corrida de datos.

Los datos que se utilizaron por default, son las coordenadas de entrada (X, Y, tomadas directamente del modelo hecho en Solidworks), las capas, neuronas, épocas y rendimiento como se muestra en la siguiente tabla:

	A	B	C	D	E	F
1	rpotOx	rpotOy	Layer	Neuron	Epoch	Performance
2	43.04	78.801	2	3	1000	0.001
3	41.28	78.802	2	3	1000	0.0001
4	39.87	78.803	2	3	1000	0.00001
5	37.94	78.804	2	4	1000	0.001
6	35.68	78.805	2	4	1000	0.0001
7	32.97	78.806	2	4	1000	0.00001
8	30.93	78.807	2	5	1000	0.001
9	28.59	78.808	2	5	1000	0.0001
10	27.1	78.809	2	5	1000	0.00001
11	24.16	78.81	2	3	2000	0.001
12	22.64	78.811	2	3	2000	0.0001
13	20.7	78.812	2	3	2000	0.00001
14	19.45	78.813	2	4	2000	0.001
15	17.82	78.814	2	4	2000	0.0001
16	16.68	78.815	2	4	2000	0.00001
17	14.87	78.816	2	5	2000	0.001
18	13.16	78.817	2	5	2000	0.0001
19	11.6	78.818	2	5	2000	0.00001
20	10.58	78.819	2	3	3000	0.001

Tabla 7. Tabla de parámetros durante el entrenamiento del modelo representativo, para 20 datos.

10) Se debe seleccionar el proceso de validación estadística de la red (p.e. Análisis de regresión).

Análisis de regresión. La cual se encarga de relacionar las variables de forma estadística.

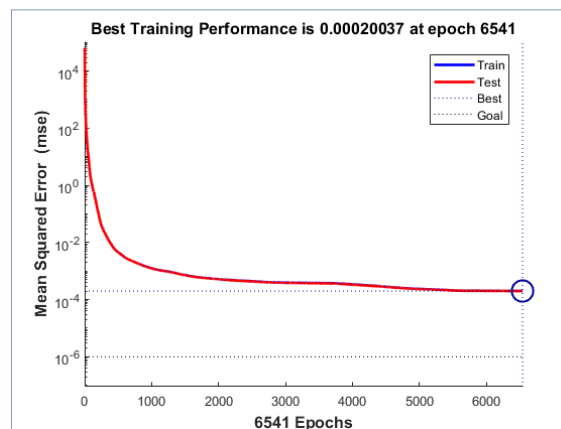


Figura 36. Rendimiento red neuronal.

La siguiente figura (figura 37) muestra el estado de entrenamiento de la red, la cual gráfica el target contra la salida (Target vs Output).

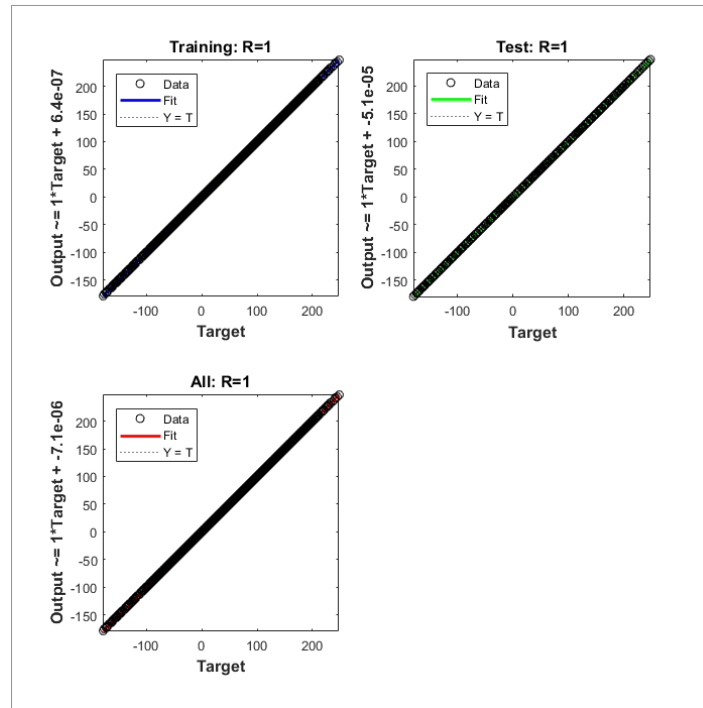


Figura 37. Estado de entrenamiento de la red neuronal.

La figura 38. Muestra el error de histograma de la red, donde se puede apreciar las instancias contra el target menos la salida, así como el entrenamiento, la prueba y el error cuando es cero.

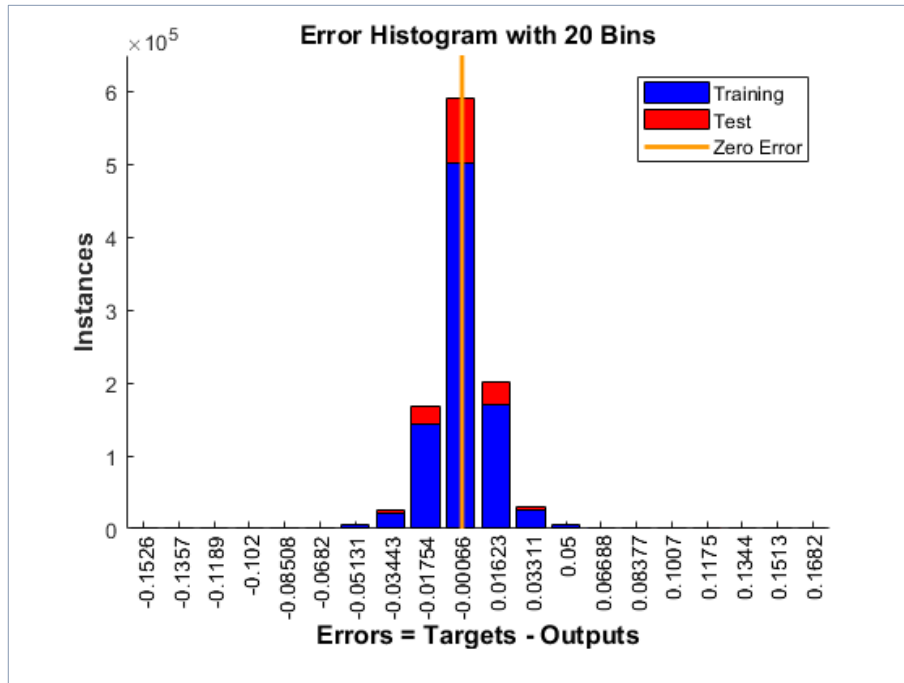


Figura 38. Error de Histograma de la red neuronal.

En la figura 39. Muestra el estado de regresión la cual se compone del gradiente, mu, número de parámetros, la suma cuadrada de los parámetros y la validación.

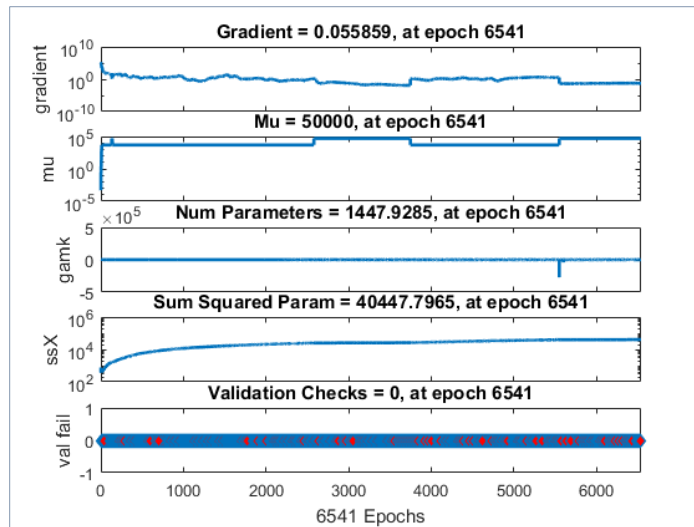


Figura 39. Estado de Regresión de la red neuronal.

11) Se debe evitar que la red se sobreentrene.

El sobreentrenamiento hace referencia al entrenamiento excesivo de la red, creando así resultados adecuados, pero con un esfuerzo mayor y más complejo.

12) Una vez entrenada la red, esta debe validarse. Puede ser de dos maneras: a) Usando la base de datos que no se entrenó o b) Diseñando una trayectoria sobre los puntos entrenados).

Para validar el entrenamiento de la red se usaron 3 datos que no se entrenaron inicialmente en la red, posteriormente se ingresaron en la red para verificar que los resultados fuesen parecidos, para ello se calculó el error dándonos así los datos que se observan en la tabla 8, teniendo un resultado satisfactorio, pues el error fue pequeño entre un método y otro.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Neuronal Network													
2	Entrada X	Entrada Y	phi1	phi2	phi3	phi4	P0	P1	Q0	Q1	R0	R1	S0	S1
3	7.37	78.899	101.253573	71.9053898	-26.008976	-155.90293	-0.1951515	0.98077311	0.89872536	-0.4385119	0.31058701	0.95054495	-0.9128551	-0.4082838
4	35.68	84.198	90.5033978	55.8942057	-24.915488	-165.56253	-0.0087858	0.9999614	0.90693017	-0.421281	0.56072273	0.82800363	-0.9684203	-0.2493232
5	32.97	120.0999	96.0911173	54.9943703	-9.7253536	179.637133	-0.1061099	0.99435441	0.98562882	-0.1689255	0.57365692	0.81909568	-0.9999799	0.00633318
6	Neuron Raphson													
7	Entrada X	Entrada Y	phi1	phi2	phi3	phi4	P0	P1	Q0	Q1	R0	R1	S0	S1
8	7.37	78.899	101.253573	71.9053898	-26.008976	-155.90293	-0.1951515	0.98077311	0.89872536	-0.4385119	0.31058701	0.95054495	-0.9128551	-0.4082838
9	35.68	84.198	90.5033978	55.8942057	-24.915488	-165.56253	-0.0087858	0.9999614	0.90693017	-0.421281	0.56072273	0.82800363	-0.9684203	-0.2493232
10	32.97	120.0999	96.0911173	54.9943703	-9.7253536	179.637133	-0.1061099	0.99435441	0.98562882	-0.1689255	0.57365692	0.81909568	-0.9999799	0.00633318
11	Error													
12	Entrada X	Entrada Y												
13	7.37	78.899	-1.2461E-10	2.1522E-11	2.7159E-11	-1.22E-10	3.7543E-12	1.3615E-12	4.6306E-12	4.7897E-12	1.592E-12	4.4202E-12	7.7908E-13	1.731E-12
14	35.68	84.198	-2.9527E-11	-3.261E-11	-2.41E-11	1.0697E-10	2.4831E-12	2.0412E-12	-1.452E-12	4.2474E-12	4.9631E-12	7.793E-13	5.412E-14	-4.491E-12
15	32.97	120.0999	-3.7945E-11	6.324E-12	3.1759E-12	1.9016E-10	4.5456E-13	3.9788E-12	-3.891E-12	-3.773E-12	-1.364E-12	-3.426E-12	1.9141E-12	3.6321E-12

Tabla 8. Tabla de validación

Se debe generar una base datos con la información de entrada y salida del Newton Rapshon y la generada por la red.

Se realizó una base de datos con las entradas y salidas tanto del método de Newton Rapshon (Tabla 9) como por la red neuronal (Tabla 10) donde se muestran los datos de los ejes y ángulos (salidas), así como las coordenadas de X y Y iniciales (entradas).

Inverse Kinematic													
rpotOx	rpotOy	phi1	phi2	phi3	phi4	P0	P1	Q0	Q1	R0	R1	S0	S1
43.04	78.801	86.61243001	52.3271863	-27.172117	-165.52687	0.05908981	0.99825267	0.88963871	-0.456665	0.61115154	0.79151361	-0.9682649	-0.249926
41.28	78.802	87.28818539	53.4139336	-27.212093	-164.84875	0.04731242	0.99888014	0.88931988	-0.4572856	0.59602962	0.80296245	-0.9652392	-0.2613679
39.87	78.803	87.83220593	54.2749405	-27.237694	-164.32223	0.03782612	0.99928434	0.88911546	-0.457683	0.58389634	0.81182823	-0.9627967	-0.2702269
37.94	78.804	88.58072069	55.439761	-27.263603	-163.62505	0.02476856	0.99969321	0.88890841	-0.458085	0.56727238	0.82353023	-0.9594373	-0.281922
35.68	78.805	89.46325055	56.7836614	-27.280107	-162.84288	0.00936791	0.99995612	0.88877642	-0.458341	0.54780182	0.836660814	-0.9554994	-0.294993
32.97	78.806	90.53059801	58.3668852	-27.279728	-161.9527	-0.0092606	0.99995712	0.88877945	-0.4583351	0.52447809	0.85142395	-0.9508011	-0.309802
30.93	78.807	91.34106203	59.5385554	-27.264393	-161.31641	-0.0234038	0.99972609	0.88890209	-0.4580972	0.50695844	0.8619705	-0.9473021	-0.3203416
28.59	78.808	92.27844572	60.8616717	-27.230652	-160.62127	-0.0397559	0.99920942	0.88917171	-0.4575737	0.48691979	0.87344669	-0.9433459	-0.331811
27.1	78.809	92.87991981	61.692651	-27.199774	-160.19781	-0.0502429	0.99873703	0.88941818	-0.4570944	0.47420114	0.88041654	-0.9408678	-0.3387739
24.16	78.811	94.07735222	63.3069044	-27.117674	-159.40402	-0.0711032	0.99746897	0.89007224	-0.4558195	0.44921134	0.89342553	-0.9360842	-0.351776
22.64	78.811	94.70241752	64.1283243	-27.063491	-159.01539	-0.0819806	0.99663393	0.8905029	-0.4549776	0.43635704	0.8997736	-0.9336766	-0.3581172
20.7	78.812	95.50623101	65.1641288	-26.98277	-158.54001	-0.095954	0.99538577	0.89114301	-0.4537225	0.42002033	0.90751469	-0.9306733	-0.3658515
19.45	78.813	96.027964	65.824075	-26.923514	-158.24601	-0.1050138	0.99447076	0.89161178	-0.4528007	0.40953974	0.91229228	-0.9287837	-0.3706222
17.82	78.814	96.71278291	66.6761838	-26.837884	-157.87646	-0.1168923	0.9931446	0.8922875	-0.4514676	0.39592724	0.91828189	-0.9262374	-0.3766049
16.68	78.815	97.19493883	67.2664323	-26.772041	-157.62741	-0.1252456	0.99212577	0.89280573	-0.4504419	0.38644646	0.92231184	-0.9247282	-0.380628
14.87	78.816	97.96580908	68.1943499	-26.657758	-157.24705	-0.1385821	0.99035094	0.89370241	-0.4486602	0.3714594	0.9284492	-0.922181	-0.3867585
13.16	78.817	98.70045206	69.0606545	-26.538246	-156.9047	-0.1512686	0.98849269	0.89463632	-0.4467951	0.35737944	0.93395928	-0.9198537	-0.3922617
11.6	78.818	99.37622876	69.8423874	-26.419204	-156.6065	-0.1629166	0.98663984	0.89556268	-0.4449354	0.34460381	0.93874822	-0.9177997	-0.3970438
10.58	78.819	99.82110191	70.3491156	-26.335963	-156.41883	-0.1705724	0.98534514	0.89620815	-0.4436338	0.33628808	0.94175917	-0.9164943	-0.4000478

Tabla 9. Entradas y salidas con método Newton Rapshon, muestra de 20 puntos.

Neuronal Network													
rpotOx	rpotOy	phi1	phi2	phi3	phi4	P0	P1	Q0	Q1	R0	R1	S0	S1
43.04	78.801	86.61243001	52.3271863	-27.172117	-165.52687	0.05908981	0.99825267	0.88963871	-0.456665	0.61115154	0.79151361	-0.9682649	-0.249926
41.28	78.802	87.28818539	53.4139337	-27.212093	-164.84875	0.04731243	0.99888014	0.88931988	-0.4572856	0.59602962	0.80296245	-0.9652392	-0.2613679
39.87	78.803	87.83220593	54.2749405	-27.237694	-164.32223	0.03782612	0.99928434	0.88911546	-0.457683	0.58389634	0.81182823	-0.9627967	-0.2702269
37.94	78.804	88.58072069	55.439761	-27.263603	-163.62505	0.02476856	0.99969321	0.88890841	-0.458085	0.56727238	0.82353023	-0.9594373	-0.281922
35.68	78.805	89.46325055	56.7836614	-27.280107	-162.84288	0.00936791	0.99995612	0.88877642	-0.458341	0.54780182	0.836660814	-0.9554994	-0.294993
32.97	78.806	90.53059801	58.3668852	-27.279728	-161.9527	-0.0092606	0.99995712	0.88877945	-0.4583351	0.52447809	0.85142395	-0.9508011	-0.309802
30.93	78.807	91.34106203	59.5385554	-27.264393	-161.31641	-0.0234038	0.99972609	0.88890209	-0.4580972	0.50695844	0.8619705	-0.9473021	-0.3203416
28.59	78.808	92.27844572	60.8616717	-27.230652	-160.62127	-0.0397559	0.99920942	0.88917171	-0.4575737	0.48691979	0.87344669	-0.9433459	-0.331811
27.1	78.809	92.87991981	61.692651	-27.199774	-160.19781	-0.0502429	0.99873703	0.88941818	-0.4570944	0.47420114	0.88041654	-0.9408678	-0.3387739
24.16	78.811	94.07735222	63.3069044	-27.117674	-159.40402	-0.0711032	0.99746897	0.89007224	-0.4558195	0.44921134	0.89342553	-0.9360842	-0.351776
22.64	78.811	94.70241752	64.1283243	-27.063491	-159.01539	-0.0819806	0.99663393	0.8905029	-0.4549776	0.43635704	0.8997736	-0.9336766	-0.3581172
20.7	78.812	95.50623101	65.1641288	-26.98277	-158.54001	-0.095954	0.99538577	0.89114301	-0.4537225	0.42002033	0.90751469	-0.9306733	-0.3658515
19.45	78.813	96.027964	65.824075	-26.923514	-158.24601	-0.1050138	0.99447076	0.89161178	-0.4528007	0.40953974	0.91229228	-0.9287837	-0.3706222
17.82	78.814	96.71278291	66.6761838	-26.837884	-157.87646	-0.1168923	0.9931446	0.8922875	-0.4514676	0.39592724	0.91828189	-0.9262374	-0.3766049
16.68	78.815	97.19493883	67.2664323	-26.772041	-157.62741	-0.1252456	0.99212577	0.89280573	-0.4504419	0.38644646	0.92231184	-0.9247282	-0.380628
14.87	78.816	97.96580908	68.1943499	-26.657758	-157.24705	-0.1385821	0.99035094	0.89370241	-0.4486602	0.3714594	0.9284492	-0.922181	-0.3867585
13.16	78.817	98.70045206	69.0606545	-26.538246	-156.9047	-0.1512686	0.98849269	0.89463632	-0.4467951	0.35737944	0.93395928	-0.9198537	-0.3922617
11.6	78.818	99.37622876	69.8423874	-26.419204	-156.6065	-0.1629166	0.98663984	0.89556268	-0.4449354	0.34460381	0.93874822	-0.9177997	-0.3970438
10.58	78.819	99.82110191	70.3491156	-26.335963	-156.41883	-0.1705724	0.98534514	0.89620815	-0.4436338	0.33628808	0.94175917	-0.9164943	-0.4000478

Tabla 10. Entradas y salidas con red neuronal, muestra de 20 puntos.

Se deben evaluar las diferencias entre los datos calculados por el Newton-Rapshon y los calculados por la red entrenada.

Se generó una base de datos (Tabla 11) donde se evalúa la diferencia de los datos calculados por el método de Neuton Rapshon y la red neuronal.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
44														
45														
46	rpotOx	rpotOy	phi1	phi2	phi3	phi4	P0	P1	Q0	Q1	R0	R1	S0	S1
47	43.04	78.801	4.64121E-11	9.513E-12	-1.307E-11	1.6801E-10	3.6044E-12	-3.651E-12	-2.367E-13	2.3106E-12	-2.461E-12	3.1656E-12	1.8428E-12	-4.506E-12
48	41.28	78.802	-4.6347E-11	-3.782E-12	4.104E-11	1.8519E-10	-2.802E-12	1.9057E-12	-3.143E-12	-4.724E-12	-2.895E-12	2.2249E-12	2.9634E-12	4.9573E-12
49	39.87	78.803	4.6833E-11	2.8058E-11	-1.408E-11	1.9411E-10	-2.318E-12	3.2004E-12	-1.214E-13	2.504E-12	-4.749E-12	7.2731E-13	-1.677E-12	4.9772E-12
50	37.94	78.804	-2.9741E-11	-2.417E-11	2.8751E-11	-2.139E-10	-1.929E-12	6.4048E-13	1.379E-12	3.5863E-12	-4.271E-13	-3.756E-12	-4.109E-12	-2.018E-12
51	35.68	78.805	-2.3898E-11	-3.498E-11	2.2563E-11	9.293E-11	1.7938E-12	1.8545E-12	-2.825E-12	8.9462E-13	-1.782E-12	7.7529E-13	-3.236E-12	1.9739E-12
52	32.97	78.806	3.33429E-11	6.032E-12	2.165E-12	4.7995E-10	2.8594E-13	1.8777E-12	3.1505E-12	-6.063E-13	6.2167E-13	4.4893E-12	-1.171E-12	-3.218E-12
53	30.93	78.807	1.10279E-11	2.2971E-11	2.2931E-11	-4.794E-10	1.3807E-12	3.0613E-12	-3.511E-12	-3.267E-12	-1.438E-12	3.0876E-12	-4.679E-12	-3.294E-12
54	28.59	78.808	-3.3493E-11	-1.194E-11	6.929E-12	3.5085E-10	4.4759E-12	-2.816E-12	1.1255E-12	1.1324E-12	-2.469E-12	1.2785E-12	4.0845E-12	-2.962E-12
55	27.1	78.809	6.6089E-12	2.3828E-11	-6.39E-12	3.012E-10	-3.794E-12	-7.462E-13	1.114E-12	-2.415E-12	-4.324E-12	-1.151E-12	-2.343E-12	-2.01E-12
56	24.16	78.81	2.2373E-11	-2.353E-11	2.725E-11	1.096E-10	-7.012E-13	3.9523E-12	2.3507E-12	1.4085E-12	-2.698E-12	4.7623E-12	3.6075E-13	5.985E-13
57	22.64	78.811	-4.7595E-11	-3.361E-11	-3.804E-11	-2.826E-10	-2.77E-12	-3.498E-12	4.6756E-13	-2.619E-12	4.5683E-12	4.3909E-13	-8.569E-13	3.4441E-12
58	20.7	78.812	-1.6909E-11	3.094E-11	-4.207E-11	1.6806E-10	1.069E-12	1.01E-12	3.2953E-12	1.3955E-12	-4.693E-12	-2.232E-12	-4.929E-12	4.6377E-12
59	19.45	78.813	2.2337E-11	1.2091E-11	3.6898E-11	-1.043E-10	-4.455E-13	3.4898E-12	1.8888E-12	-2.562E-12	3.843E-12	-4.554E-12	3.6141E-12	-4.519E-12
60	17.82	78.814	2.9853E-11	-2.29E-11	2.5692E-11	8.534E-11	-4.586E-12	1.2224E-12	4.62E-12	-1.772E-12	3.9507E-13	1.5003E-12	-4.347E-12	-2.05E-12
61	16.68	78.815	-5.456E-12	-3.481E-11	-3.177E-11	7.534E-11	4.7391E-12	-3.591E-12	-1.686E-13	1.5634E-12	1.5622E-12	-2.498E-12	-9.78E-13	4.7009E-13
62	14.87	78.816	2.874E-12	1.5124E-11	3.9898E-11	4.1716E-10	4.3536E-12	1.4778E-12	4.7873E-12	-4.983E-12	1.619E-12	4.1215E-12	3.5356E-12	2.3734E-12
63	13.16	78.817	-8.0901E-12	-2.442E-11	9.977E-12	1.0157E-10	-3.815E-12	3.4195E-12	-2.402E-13	1.5977E-12	3.6852E-12	3.0476E-13	-1.726E-12	4.7124E-12
64	11.6	78.818	3.69519E-11	-2.378E-11	8.273E-12	1.4661E-10	1.1739E-12	4.0619E-12	-7.426E-14	-5.224E-13	8.8457E-13	2.8243E-12	-3.771E-12	7.1526E-13
65	10.58	78.819	3.42621E-11	-1.639E-11	-2.037E-11	-2.188E-10	-2.418E-12	-1.427E-12	7.3436E-13	3.4889E-12	-2.629E-12	-2.81E-12	-3.041E-12	4.8717E-12

Tabla 11. Error entre red Neuronal y método de Newton Rapshon.

13) Se deben graficar los resultados.

Se creó una gráfica donde se observa mejor la tendencia de error entre los ejes y ángulos de ambos métodos como se ve desde la figura 40 hasta la figura 51.

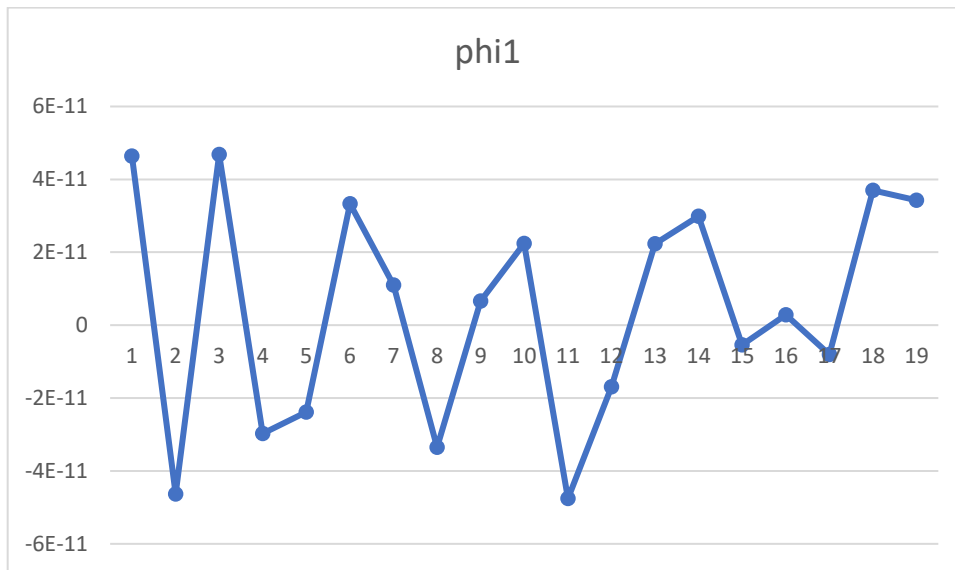


Figura 40. Error entre Red Neuronal y Newton Rapshon (phi1).

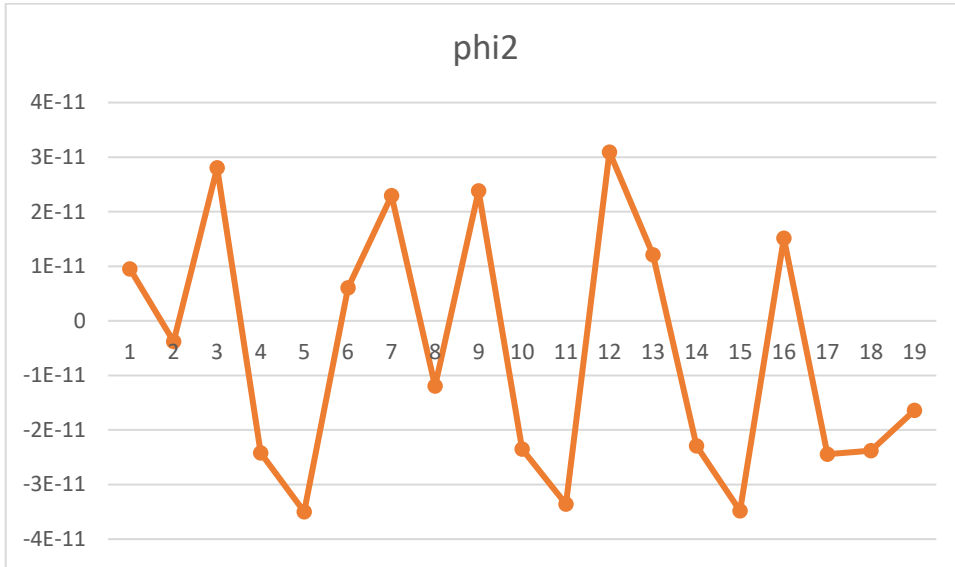


Figura 41. Error entre Red Neuronal y Newton Rapshon (ph2).

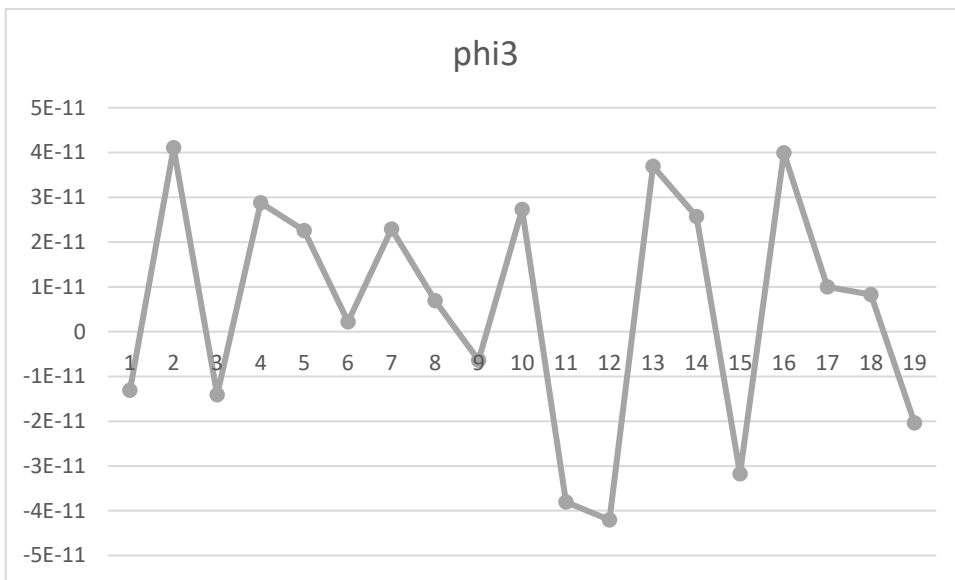


Figura 42. Error entre Red Neuronal y Newton Rapshon (ph3).

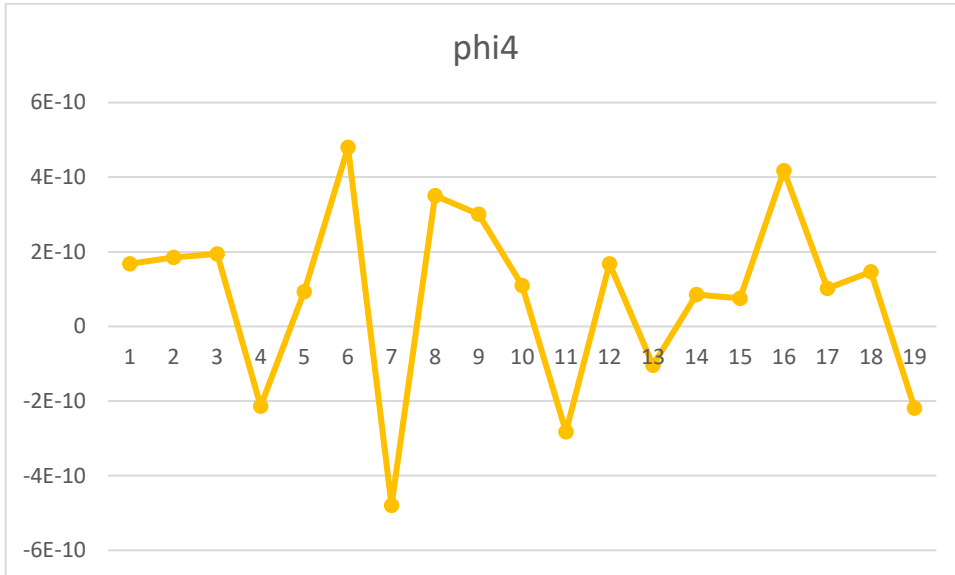


Figura 43. Error entre Red Neuronal y Newton Rapshon (ph4).

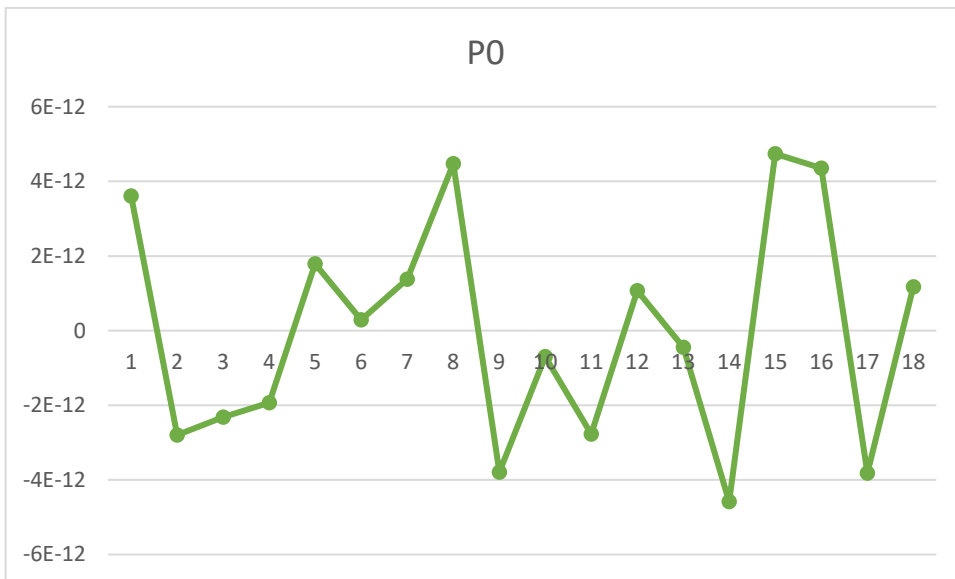


Figura 44. Error entre Red Neuronal y Newton Rapshon (P0).

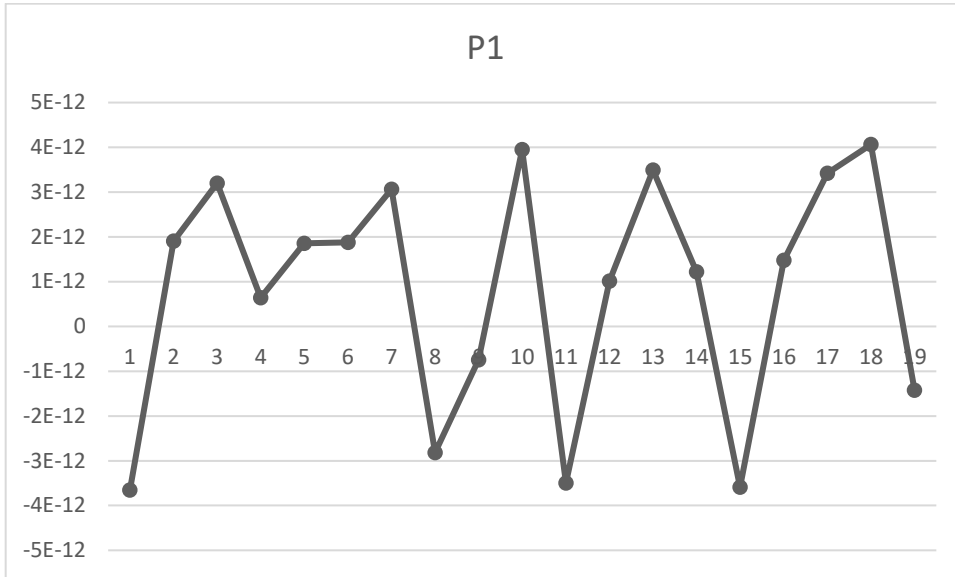


Figura 45. Error entre Red Neuronal y Newton Rapshon (P1).

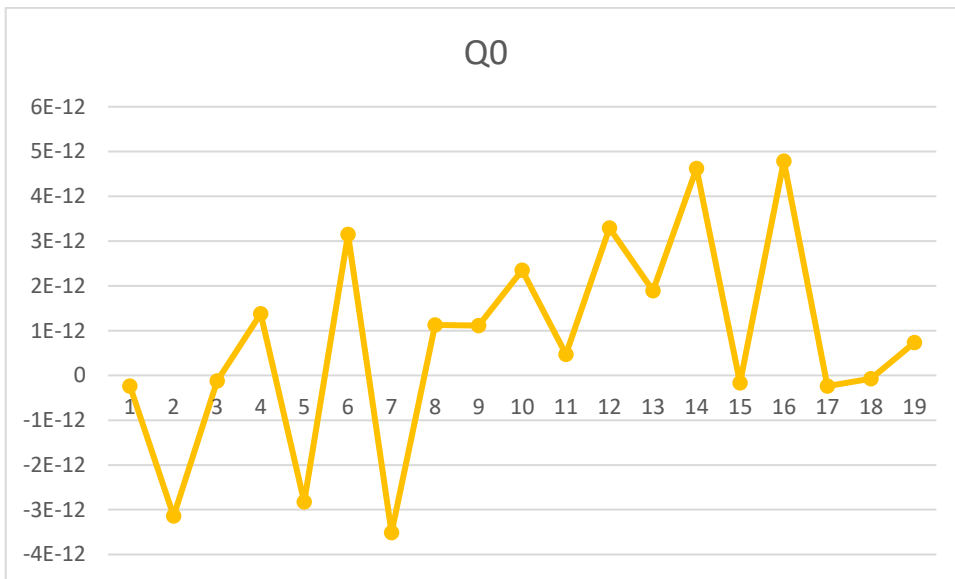


Figura 46. Error entre Red Neuronal y Newton Rapshon (Q0).

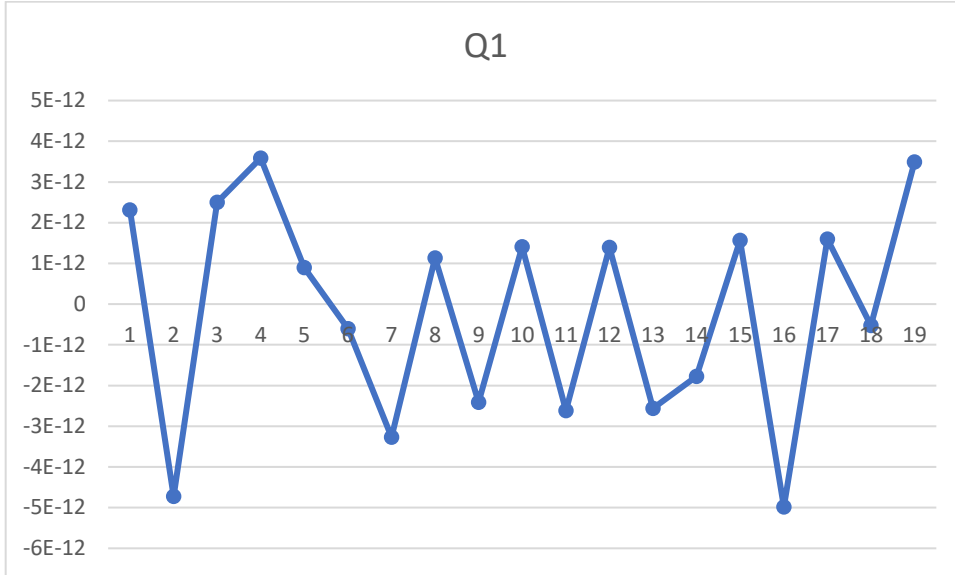


Figura 47. Error entre Red Neuronal y Newton Rapshon (Q1).

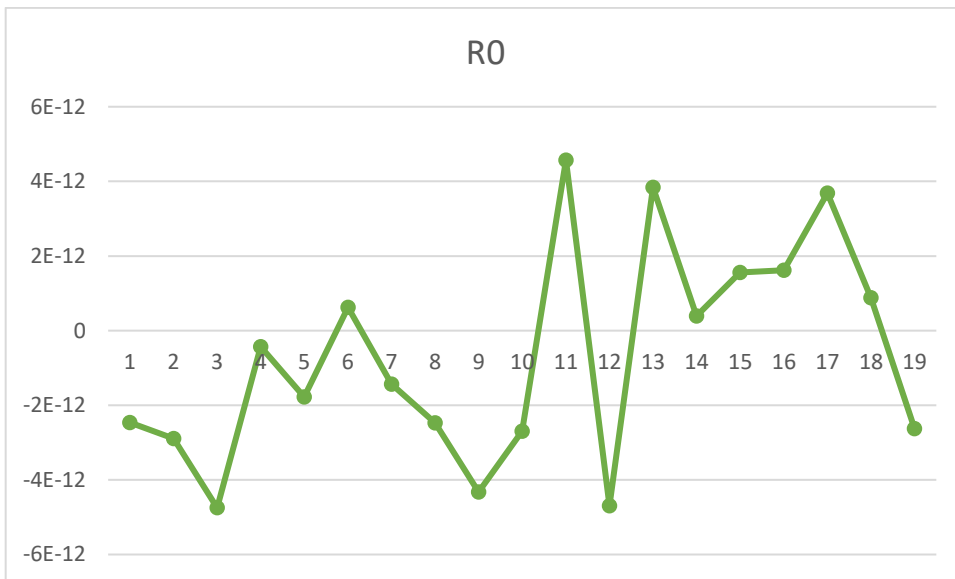


Figura 48. Error entre Red Neuronal y Newton Rapshon (R0).

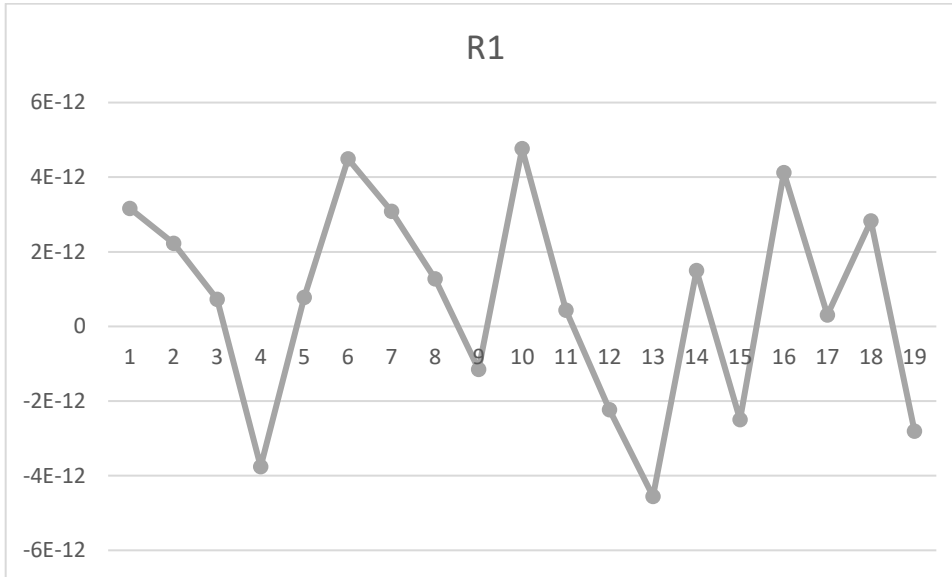


Figura 49. Error entre Red Neuronal y Newton Rapshon (R1).

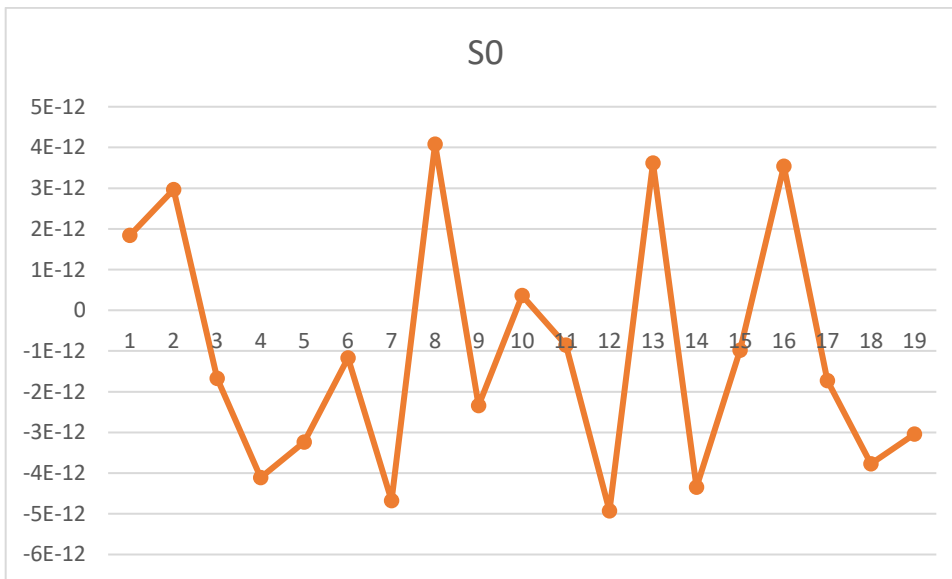


Figura 50. Error entre Red Neuronal y Newton Rapshon (S0).

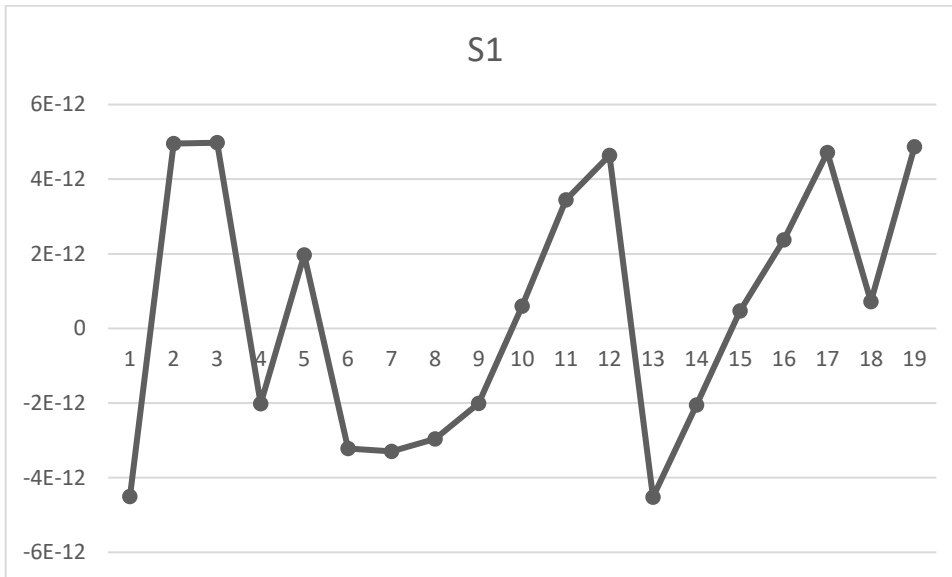


Figura 51. Error entre Red Neuronal y Newton Rapshon (S1).

14) Se deben escribir las conclusiones.

Al realizar los experimentos utilizando inicialmente el método de Newton Raphson y posteriormente la red Neuronal con un entrenamiento de Regulación Bayesiana, se pudo observar que al utilizar la red Neuronal ésta se comportaba de forma más eficiente, es decir que el resultado tiende a tener un mejor rendimiento así como una respuesta mucho más rápida en comparación a las demás, que con el método de Newton-Raphson, también se mostraba mucho más estable y constante, además los resultados que se obtuvieron no tenían una diferencia tan considerable entre ambos métodos

#### 4.2.2 Experimentación para seleccionar una topología especial.

En esta sección se presenta el desarrollo de una experimentación relacionada con el análisis de una topología de una red neuronal especial. Se desarrolla el sistema de la cinemática

inversa correspondiente al modelo representativo del modelo matemático con el que se puedan identificar los datos tanto de entrada como de salida de la base de datos que se utilizará para el entrenamiento de la red de inteligencia artificial conservando la calidad y cantidad de las diferentes combinaciones en el espacio de movimiento X y Y.

### 15) Experimentación

#### 1) Selección del objetivo a mejorar y optimizar

El interés de este experimento es determinar una red mejorada a partir de un árbol de combinaciones que se forma por la selección de varios parámetros los cuales variarán dentro de un determinado rango o intervalo. En el problema se describe como sigue:

“Dado un conjunto de soluciones de redes entrenadas, determinar una red neuronal mejorada, tal que se cumplan las condiciones siguientes:

- 1)  $G < G'$
- 2)  $E < E'$
- 3)  $A \sim A'$
- 4)  $T_0 = T'_0$

Aquí,

1. G es gradiente de entrenamiento realizado y  $G'$  un intervalo.

2. E es el menor error de Histograma registrado con respecto a la media " $\mu$ " de datos de entrada  $E/\mu$ .
3. A es el ajuste de la regresión y  $A' \sim 1.T$  siendo T el Target.
4.  $T_0$  es la tolerancia en el cálculo del modelo matemático y  $T'_0$  el valor numérico de la tolerancia.

## 2) Formulación de la hipótesis

Dado un conjunto de soluciones de redes entrenadas conformadas de un árbol de combinaciones de una selección finita de parámetros que pueden variar, existe cuando menos una topología mejorada que satisface:

- a)  $G < 0.5$
- b)  $E < \mu$
- c)  $A = T.1$
- d)  $T_0 = \pm 1.5$

## 3) Selección de las variables a manipular

Las variables asociadas con cada topología que se tomaron en cuenta para realizar el experimento fueron las siguientes:

- a) Número de Capas: 2, 3, 4.
- b) Número de Neuronas por capa: 3, 4, 5.
- c) Número de Épocas: 1000, 2000, 3000.
- d) Rendimientos: .001, .0001, .00001.

4) Descripción de las variables fijas

Las variables fijas más importantes para la experimentación fueron las siguientes:

Se tomaron 10000 datos de entrada y salida que corresponde a la siguiente configuración.

a) 2 de entrada : (rpotOx, rpotOy)

b) 12 de salida:

phi1 phi2 phi3 phi4 P0 P1 Q0 Q1 R0 R1 S0 S1

5) Definir los criterios de comparación

Los criterios de comparación serán los siguientes:

a) Comparación del error calculado

b) Comparación gráfica

6) Definir el método de validación

La validación de las pruebas se realizará considerando dos aspectos:

a) La validación interna del tool box de Matlab.

b) La validación externa usando los resultados de la red entrenada contra los datos de la base de datos del Newton-Rapshon

7) Definir el número de corridas

El número de corridas del experimento se determinó por la siguiente expresión:

$$N_c = C \times N \times E \times R$$

Siendo:

$C=3$ ; es el número de capas.

$N=3$ ; es el número de neuronas por capa.

$E=3$ ; es el número de épocas

$R=3$ ; es el número de rendimientos considerados

Por lo que:

$N_c = 81$  experimentos por las 4 variables con 3 niveles de cambios.

Se requiere entonces desarrollar 81 experimentos para determinar el conjunto de soluciones.

Para realizar la experimentación se utilizó la siguiente codificación en Matlab:

```
netbr = feedforwardnet([3 3 3], 'trainbr');
```

```
% Configuración de la red netbr con los datos |x| y |t|.
```

```
netbr = configure(netbr, Entradas', Salidas');
```

```
% Visualización de la red generada.
```

```
netbr.trainParam.epochs = 3000;    % Máximo número de épocas a entrenar (default = 1000).
```

```
netbr.trainParam.goal = 0.00001;    % Objetivo de rendimiento (default = 0).
```

```
netbr.trainParam.mu = 0.0000000001;    % Parámetro de ajuste de Marquardt (default = 0.005).
```

```
netbr.trainParam.mu_dec = 0.1;    % Factor de disminución para mu (default = 0.1).
netbr.trainParam.mu_inc = 10;    % Factor de incremento para mu (default = 10).
netbr.trainParam.mu_max = 1e10;  % Valor máximo para mu (default = 1e10).
netbr.trainParam.max_fail = 0;   % Errores máximo de validación (default = 0).
netbr.trainParam.min_grad = 1e-7; % Mínimo rendimiento de gradiente (default = 1e-
7).

netbr.inputs{1}.processFcns= {};
netbr.outputs{2}.processFcns= {};

[netbr, trbr] = train(netbr, Entradas', Salidas', 'useParallel', 'yes', 'showResources', 'yes');
```

*Cabe mencionar que diversos parámetros internos fueron tomados por default.*

8) Definir el criterio estadístico para la visualización de las corridas

Los criterios estadísticos utilizados para visualización fueron:

- a) El histograma
- b) La correlación (regresión)

9) Realizar el proceso de cómputo (entrenamiento)

La figura 52 muestra la pantalla de matlab donde se realizaron los entrenamientos.

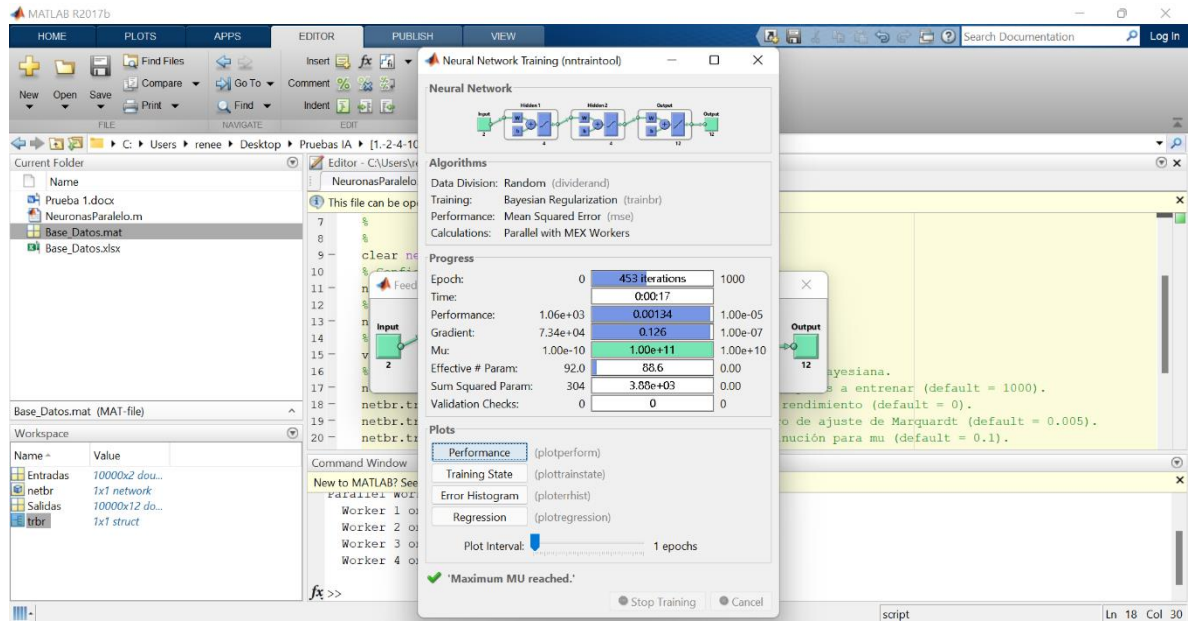


Figura 52. Pantalla frontal donde se muestra el entrenamiento y las principales variables para cada experimento.

Se observa en la Figura 52, la finalización del entrenamiento con su respectiva configuración de programación para el experimento  $[C, N, E, R]=[2, 4, 1000, 0.0000001]$ . El entrenamiento acaba cuando se cumplen las metas establecida, para la verificación del entrenamiento a lo largo de cada uno de los pasos del procedimiento.

#### 10) Análisis de la información y comprobación de la hipótesis

Una vez calculadas las 81 pruebas se procedió a sintetizar la información de proceso para su posterior filtrado (ver Tabla 11).

No. Experiment	Layer	Neuron	Epoch	Performance	Network name
1	2	3	1000	0.001	[1.-2-3-1000-0.001]
2	2	3	1000	0.0001	[1.-2-3-1000-0.0001]
3	2	3	1000	0.00001	[1.-2-3-1000-0.00001]
4	2	4	1000	0.001	[1.-2-4-1000-0.001]
5	2	4	1000	0.0001	[1.-2-4-1000-0.0001]
6	2	4	1000	0.00001	[1.-2-4-1000-0.00001]
7	2	5	1000	0.001	[1.-2-5-1000-0.001]
8	2	5	1000	0.0001	[1.-2-5-1000-0.0001]
9	2	5	1000	0.00001	[1.-2-5-1000-0.00001]
10	2	3	2000	0.001	[1.-2-3-2000-0.001]
11	2	3	2000	0.0001	[1.-2-3-2000-0.0001]
12	2	3	2000	0.00001	[1.-2-3-2000-0.00001]
13	2	4	2000	0.001	[1.-2-4-2000-0.001]
14	2	4	2000	0.0001	[1.-2-4-2000-0.0001]
15	2	4	2000	0.00001	[1.-2-4-2000-0.00001]
16	2	5	2000	0.001	[1.-2-5-2000-0.001]
17	2	5	2000	0.0001	[1.-2-5-2000-0.0001]
18	2	5	2000	0.00001	[1.-2-5-2000-0.00001]
19	2	3	3000	0.001	[1.-2-3-3000-0.001]
20	2	3	3000	0.0001	[1.-2-3-3000-0.0001]
21	2	3	3000	0.00001	[1.-2-3-3000-0.00001]
22	2	4	3000	0.001	[1.-2-4-3000-0.001]
23	2	4	3000	0.0001	[1.-2-4-3000-0.0001]
24	2	4	3000	0.00001	[1.-2-4-3000-0.00001]
25	2	5	3000	0.001	[1.-2-5-3000-0.001]
26	2	5	3000	0.0001	[1.-2-5-3000-0.0001]
27	2	5	3000	0.00001	[1.-2-5-3000-0.00001]
28	3	3	1000	0.001	[1.-3-3-1000-0.001]
29	3	3	1000	0.0001	[1.-3-3-1000-0.0001]
30	3	3	1000	0.00001	[1.-3-3-1000-0.00001]
31	3	4	1000	0.001	[1.-3-4-1000-0.001]
32	3	4	1000	0.0001	[1.-3-4-1000-0.0001]
33	3	4	1000	0.00001	[1.-3-4-1000-0.00001]
34	3	5	1000	0.001	[1.-3-5-1000-0.001]
35	3	5	1000	0.0001	[1.-3-5-1000-0.0001]
36	3	5	1000	0.00001	[1.-3-5-1000-0.00001]
37	3	3	2000	0.001	[1.-3-3-2000-0.001]
38	3	3	2000	0.0001	[1.-3-3-2000-0.0001]
39	3	3	2000	0.00001	[1.-3-3-2000-0.00001]
40	3	4	2000	0.001	[1.-3-4-2000-0.001]
41	3	4	2000	0.0001	[1.-3-4-2000-0.0001]

42	3	4	2000	0.00001	[1.-3-4-2000-0.00001]
43	3	5	2000	0.001	[1.-3-5-2000-0.001]
44	3	5	2000	0.0001	[1.-3-5-2000-0.0001]
45	3	5	2000	0.00001	[1.-3-5-2000-0.00001]
46	3	3	3000	0.001	[1.-3-3-3000-0.001]
47	3	3	3000	0.0001	[1.-3-3-3000-0.0001]
48	3	3	3000	0.00001	[1.-3-3-3000-0.00001]
49	3	4	3000	0.001	[1.-3-4-3000-0.001]
50	3	4	3000	0.0001	[1.-3-4-3000-0.0001]
51	3	4	3000	0.00001	[1.-3-4-3000-0.00001]
52	3	5	3000	0.001	[1.-3-5-3000-0.001]
53	3	5	3000	0.0001	[1.-3-5-3000-0.0001]
54	3	5	3000	0.00001	[1.-3-5-3000-0.00001]
55	4	3	1000	0.001	[1.-4-3-1000-0.001]
56	4	3	1000	0.0001	[1.-4-3-1000-0.0001]
57	4	3	1000	0.00001	[1.-4-3-1000-0.00001]
58	4	4	1000	0.001	[1.-4-4-1000-0.001]
59	4	4	1000	0.0001	[1.-4-4-1000-0.0001]
60	4	4	1000	0.00001	[1.-4-4-1000-0.00001]
61	4	5	1000	0.001	[1.-4-5-1000-0.001]
62	4	5	1000	0.0001	[1.-4-5-1000-0.0001]
63	4	5	1000	0.00001	[1.-4-5-1000-0.00001]
64	4	3	2000	0.001	[1.-4-3-2000-0.001]
65	4	3	2000	0.0001	[1.-4-3-2000-0.0001]
66	4	3	2000	0.00001	[1.-4-3-2000-0.00001]
67	4	4	2000	0.001	[1.-4-4-2000-0.001]
68	4	4	2000	0.0001	[1.-4-4-2000-0.0001]
69	4	4	2000	0.00001	[1.-4-4-2000-0.00001]
70	4	5	2000	0.001	[1.-4-5-2000-0.001]
71	4	5	2000	0.0001	[1.-4-5-2000-0.0001]
72	4	5	2000	0.00001	[1.-4-5-2000-0.00001]
73	4	3	3000	0.001	[1.-4-3-3000-0.001]
74	4	3	3000	0.0001	[1.-4-3-3000-0.0001]
75	4	3	3000	0.00001	[1.-4-3-3000-0.00001]
76	4	4	3000	0.001	[1.-4-4-3000-0.001]
77	4	4	3000	0.0001	[1.-4-4-3000-0.0001]
78	4	4	3000	0.00001	[1.-4-4-3000-0.00001]
79	4	5	3000	0.001	[1.-4-5-3000-0.001]
80	4	5	3000	0.0001	[1.-4-5-3000-0.0001]
81	4	5	3000	0.00001	[1.-4-5-3000-0.00001]

Tabla 12. Información de los 81 experimentos realizados en Matlab

Se realizaron un total de 81 experimentos variando 4 variables de aprendizaje de la IA que fueron las Capas, Neuronas, Épocas, Rendimiento para evaluar las salidas de Rendimiento, gradientes, Error, entrenamiento, Test de IA.

La Tabla 13 muestra el primer filtrado el cual está relacionado con el gradiente.

Capa	Neurona	Epoca	Rendimien	Performanc	Epoch	Grad	Epoch	ErrorHis	Training	Test
2	3	1000	0.00001	9.4269e^-06	194	0.92613	194	20	1	1*Targ+(-0.00027)
2	3	2000	0.00001	9.9456e^-06	411	0.21948	411	20	1	1*Targ+1.2e^-05
2	3	2000	0.001	0.00099763	263	0.50961	263	20	1	1*Targ+0.00055
2	4	2000	0.00001	9.7795e^-06	419	0.0041678	419	20	1	1*Targ+(-0.00041)
2	5	3000	0.00001	9.0992e^-06	493	0.4921	493	20	1	1*Targ+0.00038
3	3	2000	0.0001	7.5658e^-05	190	0.6759	190	20	1	1*Targ+0-00084
3	5	1000	0.0001	4.2738e^-05	114	0.022504	114	20	1	1*Targ+0.00059
3	5	2000	0.00001	9.7987e^-06	263	0.16689	263	20	1	1*Targ+(-0-00037
4	3	3000	0.00001	9.3807e^-06	868	0.010009	868	20	1	1*Targ+(-6.4e^-05)
4	4	3000	0.00001	9.6406e^-06	1028	0.047477	1028	20	1	1*Targ+(-7.4e^-05)
4	4	3000	0.0001	9.7517e^-05	154	0.64453	154	20	1	1*Targ+0.00085

Tabla 13. Primer filtrado en base al gradiente

Se realizó el primer filtrado con el objetivo de discriminar todos aquellos gradientes mayores a 1 que afectan la precisión del sistema al requerir un mayor ajuste para las variables de la ecuación que simulado el modelado matemático entrenando.

La Tabla 14 muestra el segundo filtrado el cual está relacionado con el rendimiento.

Capa	Neurona	Epoca	Rendimien	Performanc	Epoch	Grad	Epoch	ErrorHis	Training	Test
2	3	1000	0.00001	9.4269e^-06	194	0.92613	194	20	1	1*Targ+(-0.00027)
2	3	2000	0.00001	9.9456e^-06	411	0.21948	411	20	1	1*Targ+1.2e^-05
2	3	2000	0.001	0.00099763	263	0.50961	263	20	1	1*Targ+0.00055
2	4	2000	0.00001	9.7795e^-06	419	0.0041678	419	20	1	1*Targ+(-0.00041)
2	5	3000	0.00001	9.0992e^-06	493	0.4921	493	20	1	1*Targ+0.00038
3	3	2000	0.0001	7.5658e^-05	190	0.6759	190	20	1	1*Targ+0-00084
3	5	1000	0.0001	4.2738e^-05	114	0.022504	114	20	1	1*Targ+0.00059
3	5	2000	0.00001	9.7987e^-06	263	0.16689	263	20	1	1*Targ+(-0-00037
4	3	3000	0.00001	9.3807e^-06	868	0.010009	868	20	1	1*Targ+(-6.4e^-05)
4	4	3000	0.00001	9.6406e^-06	1028	0.047477	1028	20	1	1*Targ+(-7.4e^-05)
4	4	3000	0.0001	9.7517e^-05	154	0.64453	154	20	1	1*Targ+0.00085

Tabla 14. Primer filtrado en base al rendimiento

Se realizó el segundo filtrado con la verificación de las redes con el objetivo de tomar solo aquellas que el rendimiento fuera menos o igual a 20 con respecto a la comparación entre los datos de aprendizaje (entrada y salida) contra los datos de entrenados en la IA.

La Tabla 15 muestra el tercer filtrado el cual está relacionado con el target

Cap	Neurona	Epoca	Rendimien	Performanc	Epoch	Grad	Epoch	ErrorHis	Training	Test
2	3	1000	0.00001	9.4269e^-06	194	0.92613	194	20	1	1*Targ+(-0.00027)
2	3	2000	0.00001	9.9456e^-06	411	0.21948	411	20	1	1*Targ+1.2e^-05
2	3	2000	0.001	0.00099763	263	0.50961	263	20	1	1*Targ+0.00055
2	4	2000	0.00001	9.7795e^-06	419	0.0041678	419	20	1	1*Targ+(-0.00041)
2	5	3000	0.00001	9.0992e^-06	493	0.4921	493	20	1	1*Targ+0.00038
3	3	2000	0.0001	7.5658e^-05	190	0.6759	190	20	1	1*Targ+0.00084
3	5	1000	0.0001	4.2738e^-05	114	0.022504	114	20	1	1*Targ+0.00059
3	5	2000	0.00001	9.7987e^-06	263	0.16689	263	20	1	1*Targ+(-0.00037)
4	3	3000	0.00001	9.3807e^-06	868	0.010009	868	20	1	1*Targ+(-6.4e^-05)
4	4	3000	0.00001	9.6406e^-06	1028	0.047477	1028	20	1	1*Targ+(-7.4e^-05)
4	4	3000	0.0001	9.7517e^-05	154	0.64453	154	20	1	1*Targ+0.00085

Tabla 15. Tercer filtrado en base al target

Se llevó a cabo un cuarto filtrado con la discriminación de todas las ecuaciones que presentaran un múltiplo diferente de 1 contra el target ya que esto indica un ajuste del objetivo que se plantea llegar en la ecuación por lo que esta genera variación e inexactitud.

La Tabla 16 muestra el segundo filtrado el cual está relacionado con la suma del target

Cap	Neurona	Epoca	Rendimien	Performanc	Epoch	Grad	Epoch	ErrorHis	Training	Test
2	4	2000	0.00001	9.7795e^-06	419	0.0041678	419	20	1	1*Targ+(-0.00041)
4	3	3000	0.00001	9.3807e^-06	868	0.010009	868	20	1	1*Targ+(-6.4e^-05)
3	5	1000	0.0001	4.2738e^-05	114	0.022504	114	20	1	1*Targ+0.00059
4	4	3000	0.00001	9.6406e^-06	1028	0.047477	1028	20	1	1*Targ+(-7.4e^-05)

Tabla 16. Tercer filtrado en base al target

Se realizó un quinto filtrado con la discriminación de la ecuación de la suma del target ya que entre más cercano se encuentre el valor de suma a 0 este presentara un mayor nivel de confiabilidad y exactitud del sistema de IA al aprendido.

### Comprobación de la hipótesis:

De acuerdo con los filtrados realizados a los experimentos se determina que la red neuronal con topología [2, 4, 2000, 0.00001] mostrada en la tabla 17, es la que satisface la hipótesis planteada:

Cap.	Neurona	Epoca	Rendimien	Performanc	Epoch	Grad	Epoch	ErrorHis	Training	Test
2	4	2000	0.00001	9.7795e^-06	419	0.0041678	419	20	1	1*Targ+(-0.00041)

Tabla 17. Cuarto filtrado en base al target

La información relacionada con esta red se presenta a continuación:

La Figura 53 muestra el proceso de cálculo y entrenamiento para la red cuya topología es [2, 4, 2000, 0.00001] y la Figura 54 muestra su gráfica del rendimiento.

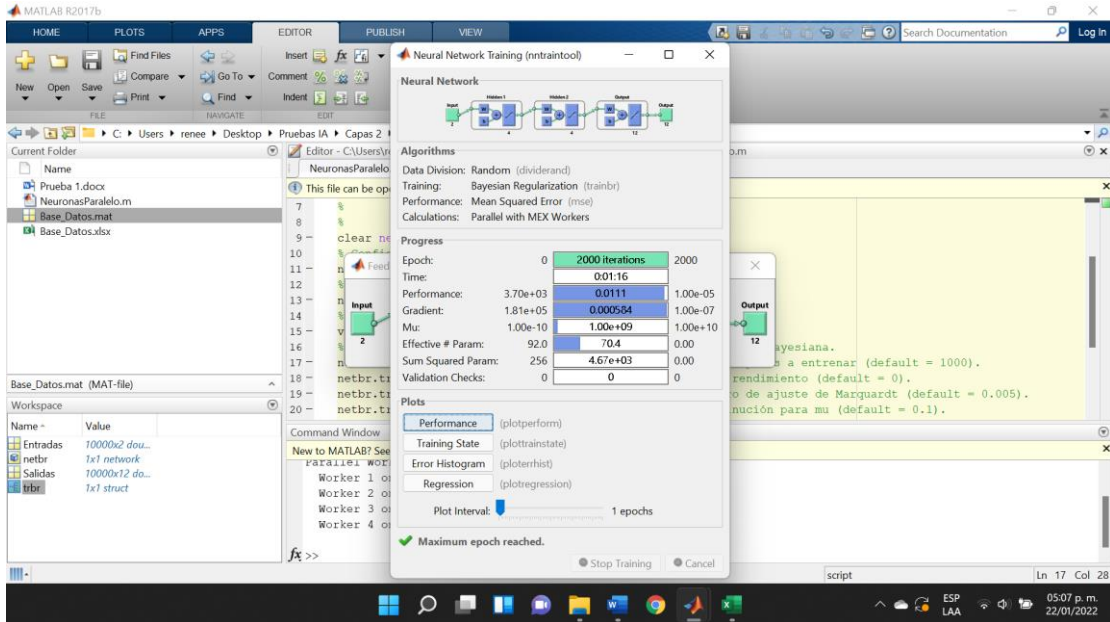


Figura 53. Entrenamiento para la red [2, 4, 2000, 0.00001].

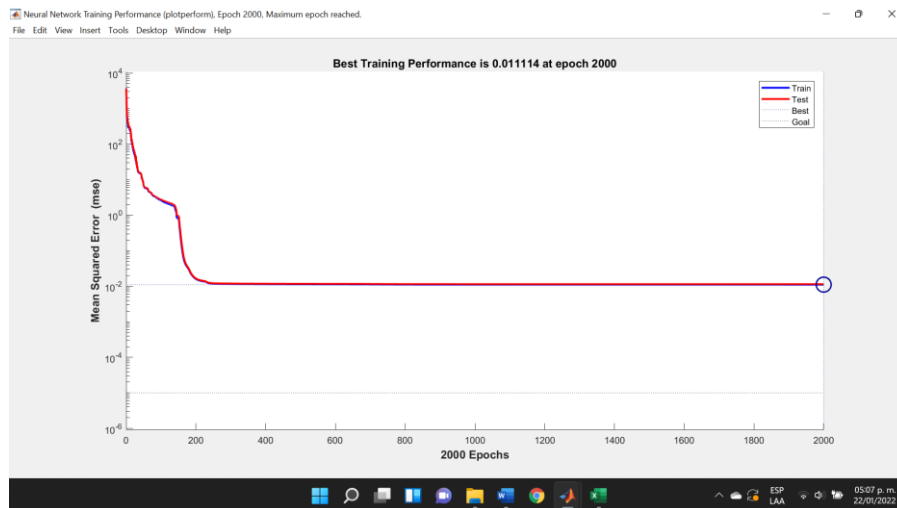


Figura 54. Gráfica del error y el número de épocas de la red [2, 4, 2000, 0.00001].

De acuerdo con la Figura anterior se puede observar la manera en que se comporta el error en cada una de las épocas comparando las entradas contra las salidas del sistema contra la base de datos al momento de realizar una interacción en el entrenamiento y la prueba del mismo.

La Figura 55 muestra algunos parámetros de la red entrenada.

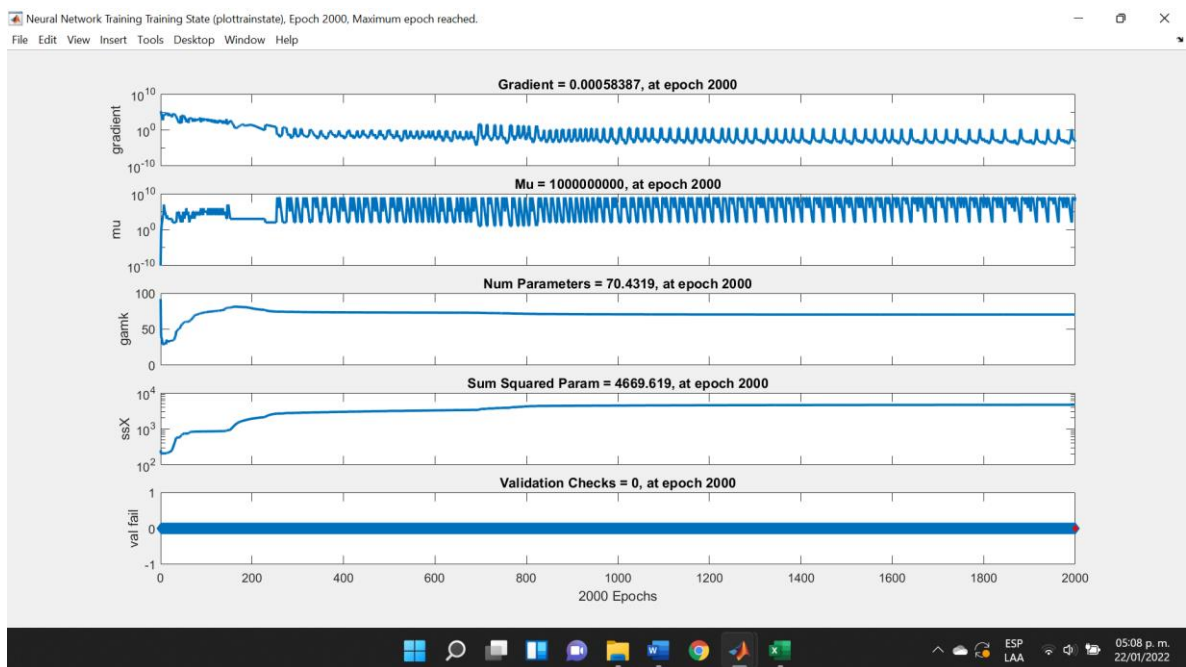


Figura 55. Parámetros de la red entrenada

La figura 55 muestra cada una de las variables y su evolución a lo largo del entrenamiento para conocer su comportamiento de control y ajuste a lo largo de todo el experimento.

La Figura 56 muestra la dispersión de los datos que deben corresponder contra el entrenamiento y la prueba de inteligencia artificial. Se observa en dicha gráfica una aproximación normal, por lo que debe generarse una forma de campana que entre más se

concentre en parte media de la campana mejor es la calidad del entrenamiento contra la base de datos que se requiere verificar.

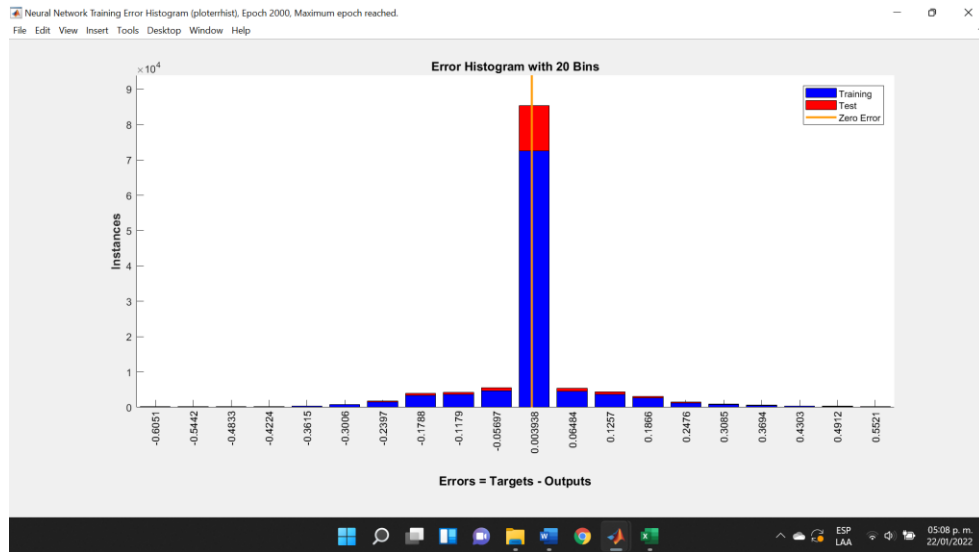


Figura 56. Histograma de errores

La Figura 57 muestra los ajustes de datos y las ganancias.

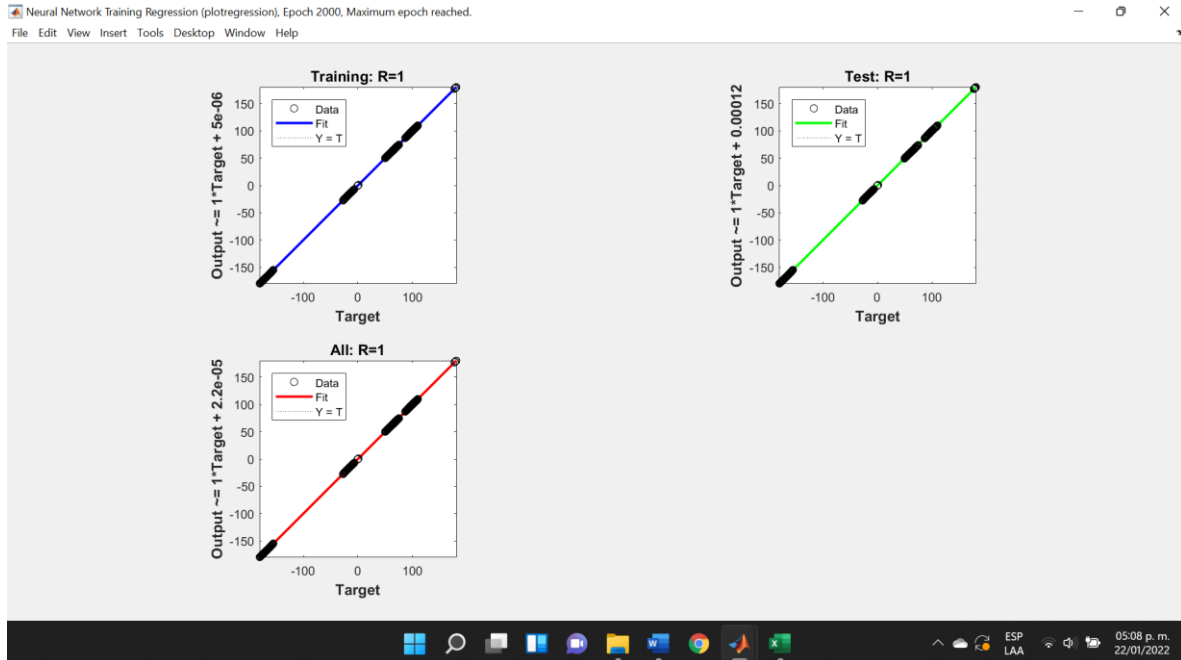


Figura 57. Ajuste de datos y ganancia

En la Figura 57 se verifica el ajuste de los datos y su respectiva ganancia, por lo que esta debe ser lo más cercano a “1” ya que quiere decir que el “target” es similar a la salida sin necesidad de un cambio de ajuste, además de que los datos son normales dentro de las regresiones que realizó el entrenamiento y se ajustó a lo largo del mismo.

La tabla 18 muestra una comparativa entre los resultados del modelo matemático resuelto con el método del Newton-Rapshon y el modelo de la Inteligencia Artificial relacionado con la configuración red [2, 4, 2000, 0.00001].

Coordenadas entrada	X	Y
		43.04
Net [4 4] Ep.2000 Ren.0.00001		
IA	MathModel	Error
86.61243001	86.61243	4.6412E-11
52.32718627	52.3271863	9.513E-12
-27.17211722	-27.1721172	-1.3067E-11
-165.526868	-165.526868	1.6801E-10
0.059089809	0.05908981	3.6044E-12
0.998252671	0.99825267	-3.6508E-12
0.889638713	0.88963871	-2.3674E-13
-0.456665042	-0.45666504	2.3106E-12
0.611151544	0.61115154	-2.4613E-12
0.791513607	0.79151361	3.1656E-12
-0.968264946	-0.96826495	1.8428E-12
-0.249925978	-0.24992598	-4.5058E-12

Tabla 18. Comparativa entre el N-R y AI.

La Tabla 18 muestra el error asociado con ambos métodos, por lo que para fines prácticos la red [2, 4, 2000, 0.00001] es una red mejorada y representativa del modelo cinemático inverso del robot en estudio.

#### 11) Descripción del mejoramiento obtenido

Con el propósito de mostrar el mejoramiento obtenido a continuación se presenta la información de la red mejorada y otra tomada de los 81 experimentos realizados. Para este caso se compararán las redes entrenadas:

- 1) [2, 4, 2000, 0.00001]
- 2) [3, 4, 2000, 0.00001]

Los resultados de la red entrenada [3, 4, 2000, 0.00001] se muestran a continuación:

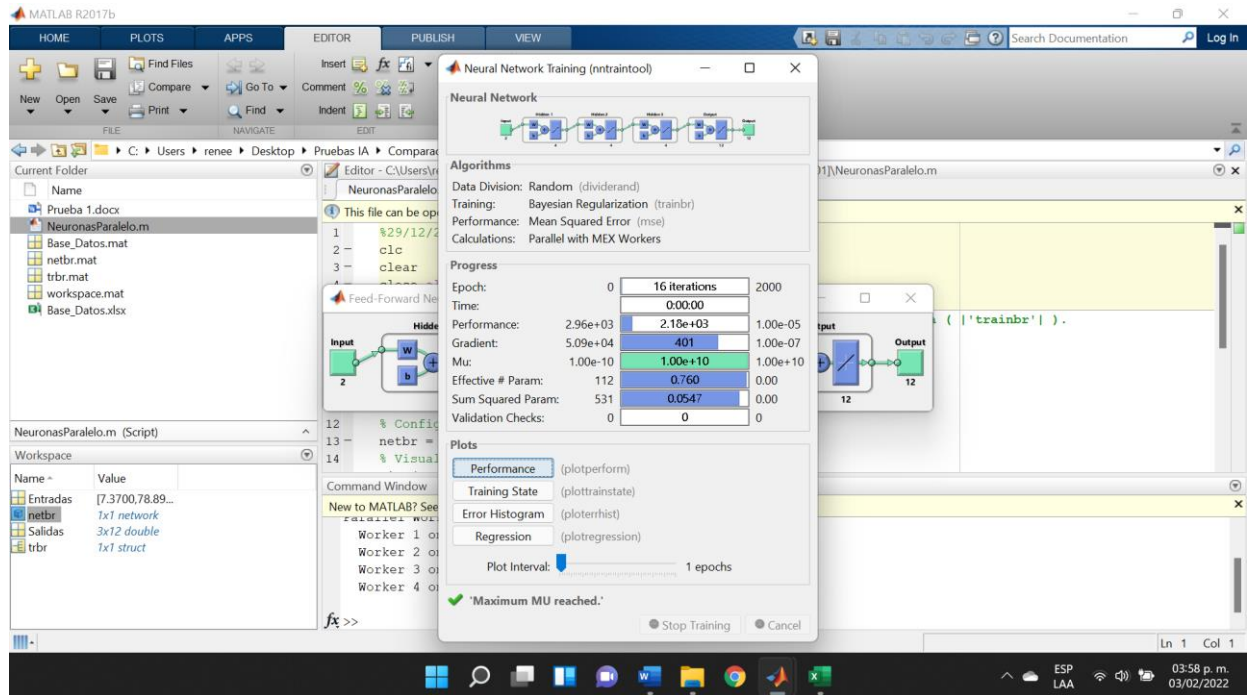


Figura 58. Entrenamiento para la red [3, 4, 2000, 0.00001].

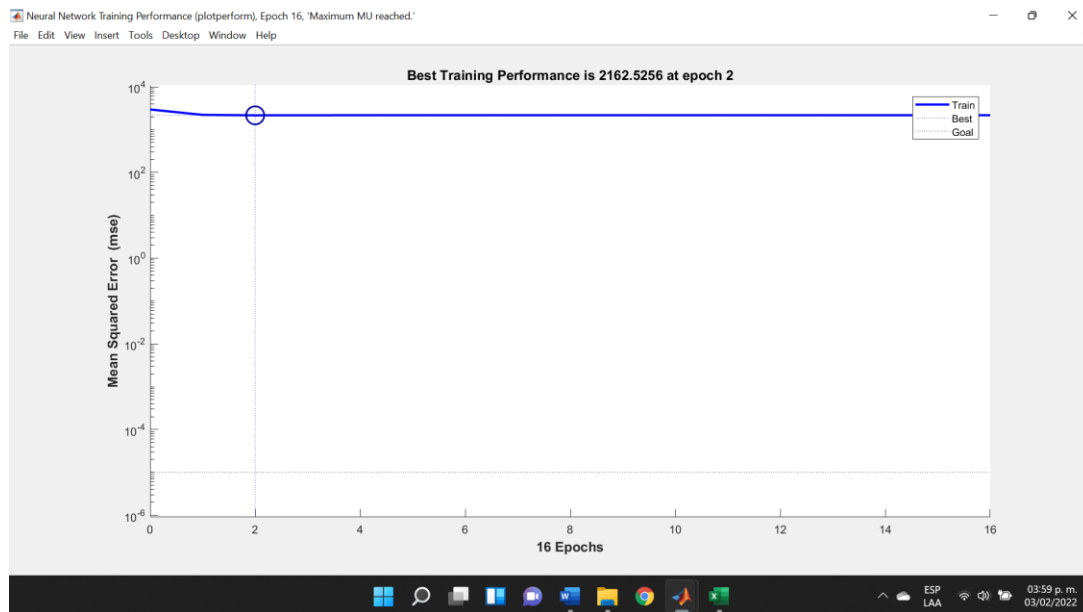


Figura 59. Gráfica del error y el número de épocas de la red [3, 4, 2000, 0.00001].

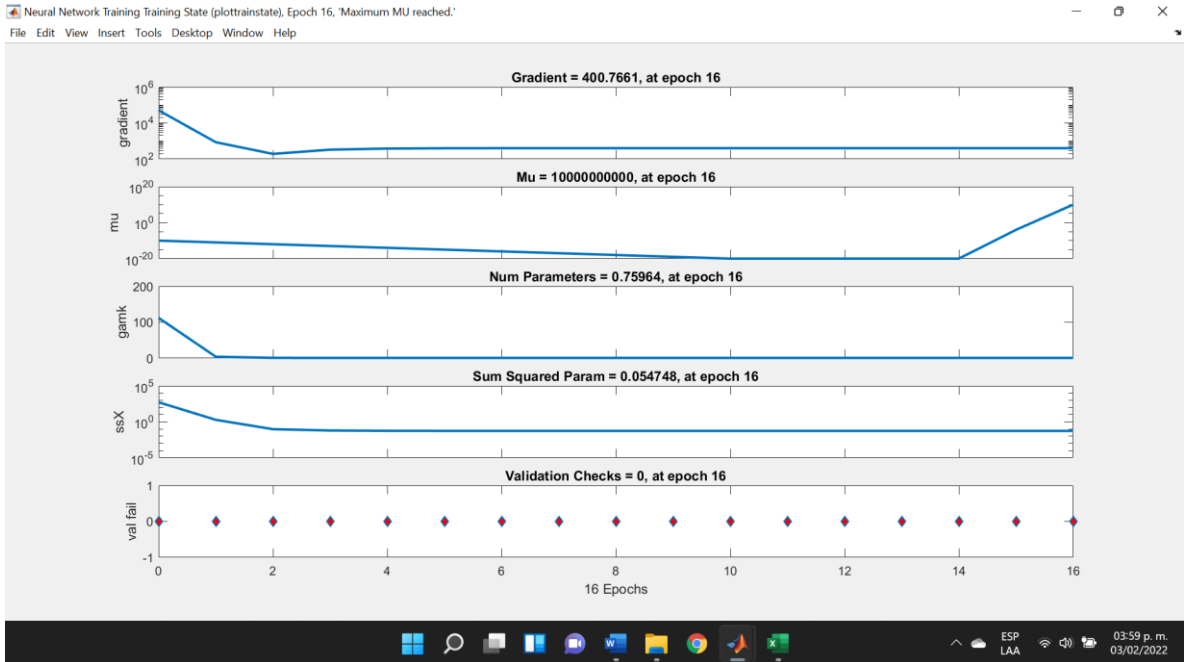


Figura 60. Parámetros de la red entrenada [3, 4, 2000, 0.00001].

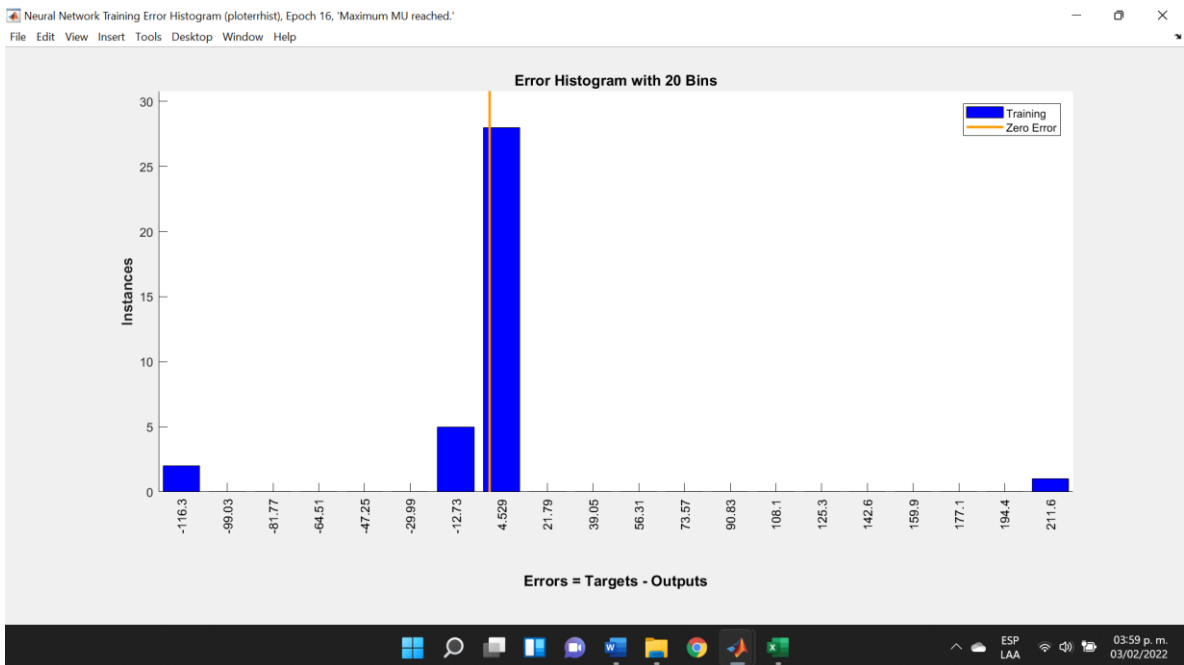


Figura 61. Histograma de errores de la red entrenada [3, 4, 2000, 0.00001].

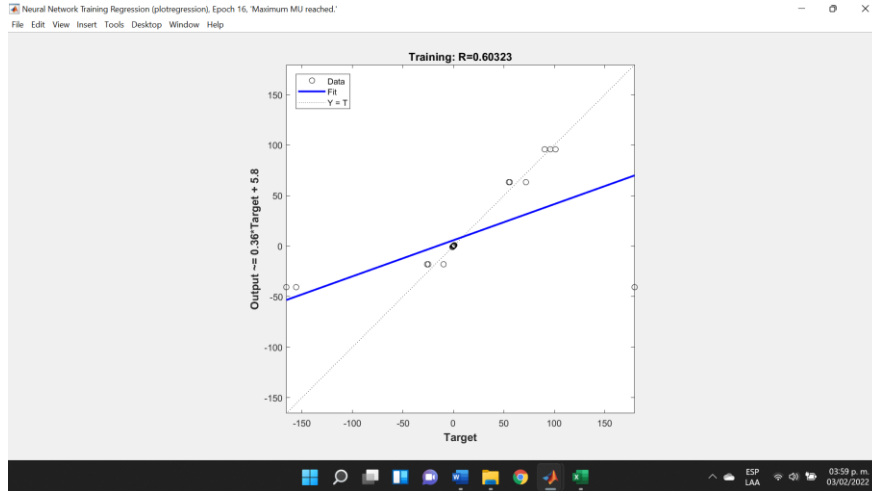


Figura 62. Ajuste de datos y ganancia de la red entrenada [3, 4, 2000, 0.00001].

Coordenadas entrada	X	Y
	43.04	78.801

IA	MathModel	Error
86.61243	86.61243	4.64121E-11
52.3271863	52.3271863	9.51303E-12
-27.1721172	-27.1721172	-1.3067E-11
-165.526868	-165.526868	1.6801E-10
0.05908981	0.05908981	3.60444E-12
0.99825267	0.99825267	-3.65082E-12
0.88963871	0.88963871	-2.3674E-13
-0.45666504	-0.45666504	2.31061E-12
0.61115154	0.61115154	-2.46134E-12
0.79151361	0.79151361	3.16556E-12
-0.96826495	-0.96826495	1.84279E-12
-0.24992598	-0.24992598	-4.50576E-12

Tabla 19. Comparativa entre el N-R y AI de la red entrenada [3, 4, 2000, 0.00001].

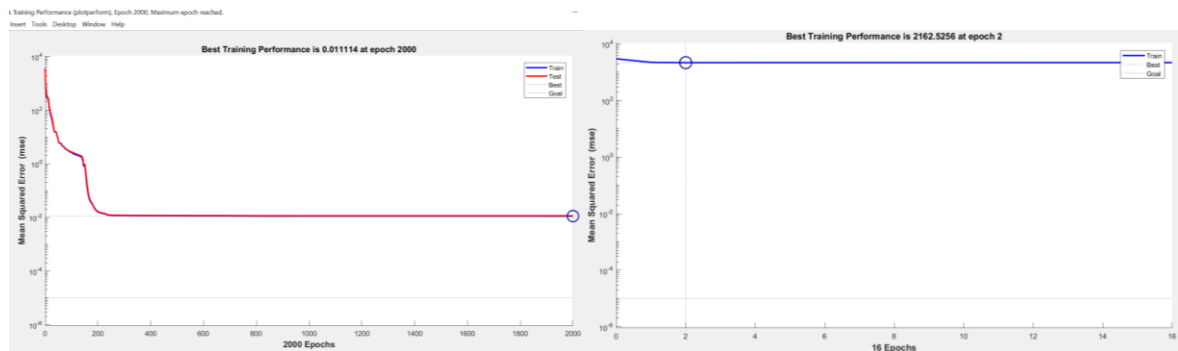
A continuación, se presentan una breve discusión sobre ambas redes [2, 4, 2000, 0.00001] (mejorada) y [3, 4, 2000, 0.00001] (entrenamiento deficiente):

Parte de los requerimientos de los aprendizajes de inteligencia artificial es cumplir los factores previamente definidos para alcanzar un mejor desempeño de procesamiento de la

información de entrada y de salida para reflejar el comportamiento del modelado matemático. Se determinó que uno de los entrenamientos de los 81 procesados se consideró como aceptable después de terminar las épocas establecidas y el otro al llegar el parámetro de ajuste de Marquardt. Si bien el sistema logró alguno de sus objetivos para detener el cálculo, se debe validar el Performance, Training State, Error Histogram, Regression y Simulación.

a) Best training Performance

En la red mejorada se puede observar que el entrenamiento contra la prueba se ajusta de forma simétrica por lo que el sistema constantemente se encuentra en ajuste buscando el mejor rendimiento posible de .011114 a 2000 épocas contra un rendimiento de 2162.5256 a 2 épocas donde no es posible comparar el entrenamiento contra la prueba (ver Figura 63).



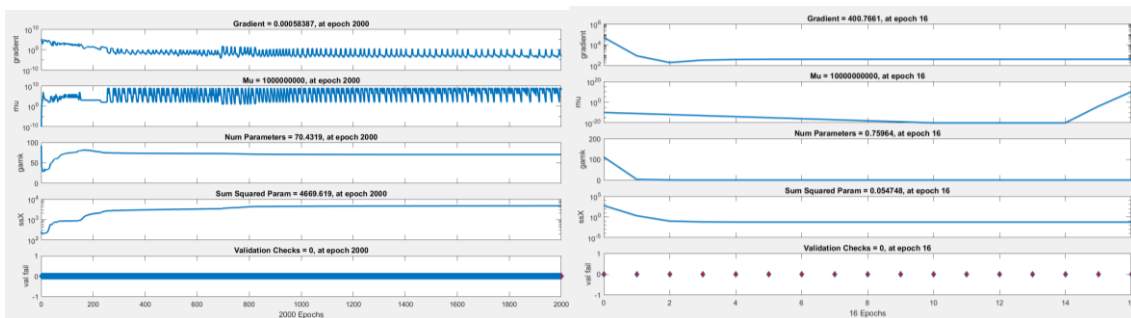
1)

2)

Figura 63. Best training Performance: 1) Red [2, 4, 2000, 0.00001] y 2) Red [3, 4, 2000, 0.00001]

### b) Training State

Continuando con el análisis del comportamiento del entrenamiento se puede observar en la figura 45 un comportamiento variable a lo largo de las 2000 épocas de forma que el ajuste es constante en la red mejorada y un comportamiento más estancando en el otro experimentó con 16 épocas, por lo que se afirma una falta de entrenamiento que no garantiza un aprendizaje mejorado (ver Figura 64).



1)

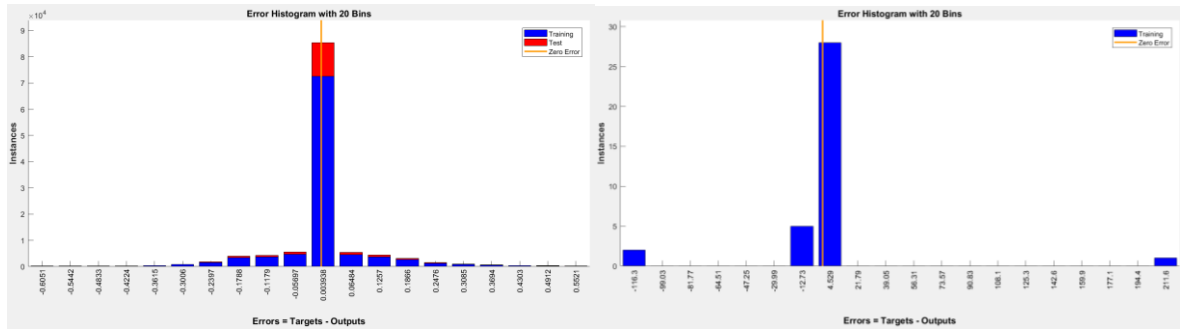
2)

Figura 64. Training State: 1) Red [2, 4, 2000, 0.00001] y 2) Red [3, 4, 2000, 0.00001]

### c) Error histogram

En el histograma del modelado mejorado (red [2, 4, 2000, 0.00001] ) se puede observar la forma ideal de una campana de los datos de entrenamiento contra la prueba lo que indica la normalidad de los datos que expresa un comportamiento similar entre las entradas y salidas de la cinemática inversa contra la inteligencia artificial, esto se debe a que la mayor parte de los datos se observan en la media del error del objetivo menos la salida de .003938

contra el error del entrenamiento no óptimo de que se ubica en un error de 4.529 además de no cumplir con el mínimo de la forma de campana (ver Figura 62).



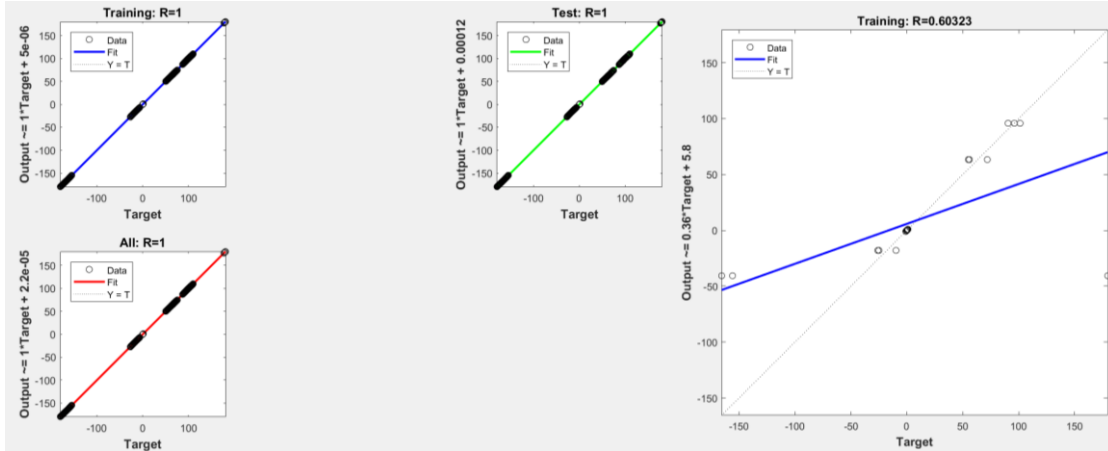
1)

2)

Figura 65. Error histogram: 1) Red [2, 4, 2000, 0.00001] y 2) Red [3, 4, 2000, 0.00001]

d) Regression

El ajuste del modelado mejorado mostrado en la Figura 47, muestra que todos los puntos se sobreponen a la línea de control que se interpreta como un ajuste de los datos de la cinemática inversa contra el de inteligencia artificial lo que resulta en su ecuación de Output = 1\*Target+5e-06, contra un entrenamiento deficiente de que no se ajuste a la línea central además de una ecuación que requiere un sobre ajuste como es Output = .36\*target + 5.8 (ver Figura 65).



1)

2)

Figura 66. Regression: 1) Red [2, 4, 2000, 0.00001] y 2) Red [3, 4, 2000, 0.00001]

## 12) Conclusiones

A modo de conclusiones se puede afirmar con los resultados obtenidos que fue posible determinar una red mejorada de un conjunto de 81 experimentos cuya naturaleza es que aprendió a calcular la cinemática inversa de un mecanismo 5R. No se puede afirmar que la red determinada es la “óptima” debido a que no se trabajó bajo un esquema o modelo de optimización formal, pero si se puede considerar que bajo los parámetros seleccionados la red [2, 4, 2000, 0.00001] fue la mejor. Por lo tanto, para las aplicaciones del robot que puedan presentarse, se recomendaría utilizar la red mejorada debido a la calidad del entrenamiento que tuvo.

## **CAPÍTULO V: CONCLUSIONES**

En este trabajo de tesis se estudió la cinemática inversa de un robot delta paralelo planar y se realizó una experimentación para determinar la topología mejorada de una red neuronal que resuelve el problema cinemático inverso. Las conclusiones que se derivan de este trabajo se resumen en los puntos siguientes:

- Se encontró una topología de red mejorada producto de modificar cuatro parámetros y de hacer una discriminación de variables. La red mejorada se obtuvo de un conjunto de 81 experimentos cuya naturaleza es que aprendió a calcular la cinemática inversa de un mecanismo 5R.

- Bajo los parámetros de discriminación la topología de la red [2, 4, 2000, 0.00001] fue la mejor.
- Para el caso de una aplicación en simulación o en un prototipo físico de un robot se recomienda utilizar la topología de la red [2, 4, 2000, 0.00001] debido a que presentó mejor ajuste lo que comprueba con las gráficas estadísticas: el histograma y la regresión.
- Para diseñar un experimento como el planteado en esta tesis, se debe generar un modelo representativo de red, generado por la base de datos y el target.
- El modelado del robot es necesario para la generación de la base de datos a entrenar.

Algunas recomendaciones asociadas con este trabajo de tesis son las siguientes:

- 1) Se deben realizar más experimentos moviendo otras variables para conocer el comportamiento de la red mejorada en caso de que exista.
- 2) Se deben utilizar modelos de optimización para obtener una verdadera topología de una red optimizada.
- 3) Se recomienda utilizar el protocolo 3 para aplicaciones en otros modelos de robot más complejos.

## ANEXO

### Anexo 1

```

%20/01/2022
% Posición articular - configuración no deformada.
% 10 puntos arbitrarios.
%           X           Y
puntos = xlsread("Base_Datos.xlsx","A2:B10001");
% componentes XY del vector rpotO.
rpotOx = puntos(:,1);
rpotOy = puntos(:,2);
rpotO = puntos';

% condiciones iniciales.
ci = [-0.9; 0; 0.8; 0.5; -0.4; 0.8; -0.9; -0.2];
error=[0,0,0,0,0,0];
% ci = -1 + (1+1)*rand(8,1);
% Variable para acumular los parámetros complejos.
parametros_ = [];

for k = 1:length(puntos)
    % Llamada a la función del problema inverso de posición.
    parametros = PosNoDefProbInv(rpotO(:,k), ci, 1e-12, 100);

    % Obtención de las posiciones en términos de parámetros complejos.
    parametros_ = [parametros_, parametros];
end

% Parámetros complejos solución.
p0 = parametros_(1,:);
p1 = parametros_(2,:);
q0 = parametros_(3,:);
q1 = parametros_(4,:);
r0 = parametros_(5,:);
r1 = parametros_(6,:);
s0 = parametros_(7,:);
s1 = parametros_(8,:);

% Definición de ángulos de posición.
phi1 = rad2deg(acos(p0).*sign(asin(p1)));
phi3 = rad2deg(acos(q0).*sign(asin(q1)));
phi2 = rad2deg(acos(r0).*sign(asin(r1)));
phi4 = rad2deg(acos(s0).*sign(asin(s1)));

% Despliegue de datos.
disp('Cinemática inversa del mecanismo 5R para 10 puntos arbitrarios')
T = table(rpotOx,rpotOy, phi1', phi2', phi3', phi4');
T.Properties.VariableNames = {'rpotOx' 'rpotOy' 'phi1' 'phi2' 'phi3' 'phi4'};
disp(T)

t=xlsread("Coordenadas red.xlsx","C2:N10001");
x=puntos;
%nnntool
%% Generación y entrenamiento de una red usando Regularización Bayesiana (
|'trainbr'| ).

```

```

clear netbr
% Configuraci3n de la red y el algoritmo de entrenamiento.
netbr = feedforwardnet([25 25 25], 'trainbr');
% Configuraci3n de la red netbr con los datos |x| y |t|.
netbr = configure(netbr, x, t);
% Visualizaci3n de la red generada. view(netbr)
% Par3metros de entrenamiento del algoritmo de Regularizaci3n Bayesiana.
netbr.trainParam.epochs = 20000; % M3ximo n3mero de 3pocas a entrenar (default
= 1000).
netbr.trainParam.goal = 1e-4; % Objetivo de rendimiento(default = 0).
netbr.trainParam.mu = 0.005; % Par3metro de ajuste de Marquardt (default =
0.005).
netbr.trainParam.mu_dec = 0.1; % Factor de disminuci3n para mu (default = 0.1).
netbr.trainParam.mu_inc = 10; % Factor de incremento para mu (default = 10).
netbr.trainParam.mu_max = 1e10; % Valor m3ximo para mu (default = 1e10).
netbr.trainParam.max_fail = 0; % Errores m3ximo de validaci3n (default = 0).
netbr.trainParam.min_grad = 1e-7; % M3nimo rendimiento de gradiente (default =
1e-7).
% Entrenamiento de la red y activaci3n del trabajo en Paralelo de los n3cleos
del procesador.
[netbr, trbr] = train(netbr, x, t, 'useParallel', 'yes', 'showResources', 'yes');

%% Funciones de soluci3n cinem3tica del cuerpo r3-gido.

% Configuraci3n no deformada - Problema inverso de posici3n.
function [ Dk1, J, it ] = PosNoDefProbInv( rpot0, Dk, e, itermaximas )
%NEWTON-RAPHSON M3todo de soluci3n num3rica.
% rpot0 es el vector desde el punto pot hasta el origen 0.
% Dk es el vector con las condiciones iniciales para p0, p1, q0, q1, r0,
% r1, s0, s1.
% e es el error dado.
% itermaximas es el m3ximo de iteraciones a realizar.

% Posici3n en el 3rgano terminal.
rpot0x = rpot0(1);
rpot0y = rpot0(2);

% Definici3n de constantes.
r10x = -95;
r10y = 0;
r30x = 95;
r30y = 0;
L21 = 145.5;
Lpot2 = 145.5;
L43 = 145.5;
Lpot4 = 145.5;

for k = 1:itermaximas
% Declaraci3n y actualizaci3n de los par3metros p0, p1, q0, q1, r0,
% r1, s0 y s1.
p0 = Dk(1);
p1 = Dk(2);
q0 = Dk(3);
q1 = Dk(4);
r0 = Dk(5);
r1 = Dk(6);
s0 = Dk(7);
s1 = Dk(8);

% Declaraci3n del vector de funciones tales que f(x) = 0.

```

```

F = [-rpot0x + r10x + L21*p0 + Lpot2*q0;
     -rpot0y + r10y + L21*p1 + Lpot2*q1;
     -rpot0x + r30x + L43*r0 + Lpot4*s0;
     -rpot0y + r30y + L43*r1 + Lpot4*s1;
     p0^2 + p1^2 - 1;
     q0^2 + q1^2 - 1;
     r0^2 + r1^2 - 1;
     s0^2 + s1^2 - 1];

% Declaraci3n de la matriz Jacobiana.
J = [ L21,    0, Lpot2,    0,    0,    0,    0,    0;
      0,    L21,    0, Lpot2,    0,    0,    0,    0;
      0,    0,    0,    0, L43,    0, Lpot4,    0;
      0,    0,    0,    0,    0, L43,    0, Lpot4;
     2*p0, 2*p1,    0,    0,    0,    0,    0,    0;
      0,    0, 2*q0, 2*q1,    0,    0,    0,    0;
      0,    0,    0,    0, 2*r0, 2*r1,    0,    0;
      0,    0,    0,    0,    0,    0, 2*s0, 2*s1];

% Se calcula D al resolver el sistema D = inv(J)*(-F).
D = J\(-F);

% Se calculan los valores de Dk1.
Dk1 = Dk + D;

% Se calcula el error con la norma de D.
ec = norm(D);

% Se pregunta si el error dado (e) es mayor que el error calculado
% (ec).
if (ec < e)
    it = k;
    break;
end

% Se asigna Dk = Dk1 para evaluar F y J con los nuevos valores de
% p0, p1, q0, q1, r0, r1, s0 y s1.
Dk = Dk1;
end
end

```

## Ap3ndice 2

```

%% Posici3n articular - configuraci3n no deformada.

% 10 puntos arbitrarios.
%           X           Y
puntos = xlsread("Base_Datos.xlsx", "A2:B10001");
%puntos = load("Prueba_1.mat");
% componentes XY del vector rpot0.
rpot0x = puntos(:,1);
rpot0y = puntos(:,2);
rpot0 = puntos';

% condiciones iniciales.

```

```

ci = [-0.9; 0; 0.8; 0.5; -0.4; 0.8; -0.9; -0.2];

eror=[0,0,0,0,0,0];
% ci = -1 + (1+i)*rand(8,1);

% Variable para acumular los parámetros complejos.
parametros_ = [];

for k = 1:length(puntos)
    % Llamada a la función del problema inverso de posición.
    parametros = PosNoDefProbInv(rpotO(:,k), ci, 1e-12, 100);

    % Obtención de las posiciones en términos de parámetros complejos.
    parametros_ = [parametros_, parametros];
end

% Parámetros complejos solución.
p0 = parametros_(1,:);
p1 = parametros_(2,:);
q0 = parametros_(3,:);
q1 = parametros_(4,:);
r0 = parametros_(5,:);
r1 = parametros_(6,:);
s0 = parametros_(7,:);
s1 = parametros_(8,:);

% Definición de ángulos de posición.
phi1 = rad2deg(acos(p0).*sign(asin(p1)));
phi3 = rad2deg(acos(q0).*sign(asin(q1)));
phi2 = rad2deg(acos(r0).*sign(asin(r1)));
phi4 = rad2deg(acos(s0).*sign(asin(s1)));

% Despliegue de datos.
disp('Cinemática inversa del mecanismo 5R para 10000 puntos arbitrarios')
T = table(rpotOx,rpotOy, phi1', phi2', phi3', phi4');
T.Properties.VariableNames = {'rpotOx' 'rpotOy' 'phi1' 'phi2' 'phi3' 'phi4'};
disp(T)

```

## REFERENCIAS

Acosta, J. A., y López, M. (2004b). *Manual de Introducción a MATLAB*. Universidad de Sevilla.

<http://www.esi2.us.es/~mlm/RAN/ManualMatlabRAN.pdf>.

Aracil, R. & Saltaren, R., y Sabater J., y Reinoso, O. (2010). Robots Paralelos: Máquinas con un Pasado para una Robótica del Futuro. *Revista iberoamericana de automatica e informatica industrial (RIAI)*, Vol. 3, Nº. 1, 2006, pags. 16-28.

Barrientos, A. (2012). *Fundamentos de robótica* (2a. ed.). Madrid, Spain: McGraw-Hill España. Recuperado de <https://elibro.net/es/ereader/ulsanoroeste/50193?page=456>.

Barros Losada, T. (2017). *La Industria 4.0: Aplicaciones e Implicaciones* (TFG). Universidad de Sevilla.

<http://bibing.us.es/proyectos/abreproy/91146/fichero/La+Industria+4.0+Aplicaciones+e+Implicaciones.pdf>

Bastién, G. M., Becerril, H. S., Díaz, G., Pérez, A. R., & Rodríguez, A. (2007). *INTRODUCCIÓN A LA DINÁMICA DEL CUERPO RÍGIDO*. Universidad Autónoma Metropolitana Unidad Azcaltotzalco. <https://core.ac.uk/download/pdf/48398555.pdf>

Becerra Espinosa, J. M. (s. f.). *Conjuntos*. Universidad Nacional Autónoma de México.

Recuperado 13 de enero de 2021, de

<http://dgenp.unam.mx/direccgral/secacad/cmatematicas/pdf/m4unidad01.pdf>

Cano, A. (2017, 10 septiembre). *Cadena cinética abierta VS Cadena cinética cerrada*. Centro impulso. <https://centroimpulso.es/cadena-cinetica/>

Cardoso, E., Fernández, A., Marrero-Osorio, S., & Guardado, P. (2017). Modelos cinemático y dinámico de un robot de cuatro grados de libertad. *Revista de Ingeniería Electrónica, Automática y Comunicaciones.*, 38, 56-75.  
<http://scielo.sld.cu/pdf/eac/v38n3/eac06317.pdf>

Carranza Bravo, P. (2010, septiembre). *INTRODUCCIÓN A LAS TÉCNICAS DE INTELIGENCIA ARTIFICIAL APLICADAS A LA GESTIÓN FINANCIERA EMPRESARIAL*. Scielo.  
[http://www.scielo.org.bo/scielo.php?script=sci\\_arttext&pid=S2071-081X2010000100002](http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S2071-081X2010000100002)

Chaillou, L. L. (2008). *CALCULO NUMERICO*. Universidad Nacional de Santiago del Estero. Facultad de agronomía y agroindustria.  
[http://faa.unse.edu.ar/pdf/publicdoc/Calculo\\_numerico.pdf](http://faa.unse.edu.ar/pdf/publicdoc/Calculo_numerico.pdf)

Chong, M. (2009). *Robótica e inteligencia artificial*. Santa Fe, Argentina, Argentina: El Cid Editor | apuntes. Recuperado de <https://elibro.net/es/ereader/ulsanoroeste/28875?page=5>.

Del Val., J. L. (2016, 18 marzo). industria 4.0 | Revista Ingeniería. *REVISTA DEUSTO INGENIERÍA*. <https://revistaingenieria.deusto.es/tag/industria-4-0/>

Díaz, M., Quintero, H. F., Mejía, L. A., Holguin, G., Herrera, M., Mesa, C., & Daraviña, G. (2018, 13 noviembre). *Aplicación de los Robots Paralelos*. Hal archives-ouvertes. <https://hal.archives-ouvertes.fr/hal-01907282/document>

Díaz M., J. (2007, 13 febrero). *Interpolación*. Universidad de Valencia.

[https://www.uv.es/diazi/cn\\_tema5.pdf](https://www.uv.es/diazi/cn_tema5.pdf)

Díaz Rodríguez, M., *Aplicación de los robots paralelos* (2018), En *Manipuladores paralelos: Síntesis, análisis y aplicaciones* (Pereira: Universidad Tecnológica de Pereira).

Díaz Medina, J. (2009). *Raíces de ecuaciones no lineales*. Universitat de València.

[https://www.uv.es/diazi/cn\\_tema2.pdf](https://www.uv.es/diazi/cn_tema2.pdf)

Fernández, J. L. (s. f.). *Trayectoria*. Fisicalab. Recuperado 6 de enero de 2021, de

<https://www.fisicalab.com/apartado/trayectoria>

Fernández, J. L. (s. f.-b). *Trayectoria y ecuación de posición*. Fisicalab. Recuperado 6 de enero de 2021, de <https://www.fisicalab.com/apartado/ecuacion-trayectoria>

Fernández, J. L. (s. f.-a). *Diferencia entre Desplazamiento y Espacio Recorrido*. FisicaLab.

Recuperado 11 de marzo de 2021, de

<https://www.fisicalab.com/apartado/desplazamiento-vs-espacio>

Fernández, J. L. (s. f.-b). *Posición*. FisicaLab. Recuperado 11 de marzo de 2021, de

<https://www.fisicalab.com/apartado/posicion>

Fernández, J. L. (s. f.-c). *¿Qué es el Movimiento?* FisicaLab. Recuperado 11 de marzo de 2021,

de <https://www.fisicalab.com/apartado/concepto-movimiento>

Fernández, J. L. (s. f.-d). *Rotación del Sólido Rígido*. FisicaLab. Recuperado 11 de marzo de

2021, de <https://www.fisicalab.com/apartado/rotacion-solido-rigido>

Figuroa, M. (2010). Física. Miami, FL, United States of America: Firms Press. Recuperado de <https://elibro.net/es/ereader/ulsanoroeste/36340?>.

García, C. A., Castellanos, E. X., & García, M. V. (2019). Desarrollo de sistemas ciber-físicos de producción para Procesamiento por lotes usando normas IEC-61499 e ISA-88. *Ingeniare. Revista chilena de ingeniería*, 27(3), 443–453. <https://scielo.conicyt.cl/pdf/ingeniare/v27n3/0718-3305-ingeniare-27-03-00443.pdf>

García Merino, F., & Ruiz Dalpivar, A. (2004). *Análisis y propuesta de un modelo logístico para la empresa Productos Agropecuarios Santa Cecilia S. A. de C. V.* (TFG). Universidad de las Américas Puebla. [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lad/garcia\\_m\\_f/](http://catarina.udlap.mx/u_dl_a/tales/documentos/lad/garcia_m_f/)

Granja, M. V. (2014, octubre). MODELACIÓN Y ANÁLISIS DE LA CINEMÁTICA DIRECTA E INVERSA DEL MANIPULADOR STANFORD DE SEIS GRADOS DE LIBERTAD. <https://bibdigital.epn.edu.ec/bitstream/15000/8693/1/CD-5831.pdf>

Hernández Sánchez, J., & Rojas Fierro, J. J. (2004). *Análisis de los factores que intervienen en la convivencia de los residentes de un colegio residencial. Caso: Ray Lindley* (TFG). Universidad de las Américas Puebla. [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lad/hernandez\\_s\\_j/](http://catarina.udlap.mx/u_dl_a/tales/documentos/lad/hernandez_s_j/)

Hinojosa Gutiérrez, Á. (2015). Python paso a paso. Paracuellos de Jarama, Madrid, RA-MA Editorial. Recuperado de <https://elibro.net/es/ereader/ulsanoroeste/107213?page=21>.

Inteligencia Artificial. (s. f.). Repositorio digital de la Facultad de Ingeniería - UNAM. Recuperado 4 de marzo de 2021, de

<http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/219/A7.pdf?sequence=7>

ISO. (2018). Conectar con colegas robóticos. *ISOfocus*, 22-27.

[https://www.iso.org/files/live/sites/isoorg/files/news/magazine/ISOfocus%20\(2013-NOW\)/sp/ISOfocus\\_131\\_sp.pdf](https://www.iso.org/files/live/sites/isoorg/files/news/magazine/ISOfocus%20(2013-NOW)/sp/ISOfocus_131_sp.pdf)

Jiménez, E., Servín, D., Reyes, L. A., Servín, R., Melendez, J., & López, A. A. (2021a). Modeling of Inverse Kinematic of 3-DoF Robot, Using Unit Quaternions and Artificial Neural Network. *Cambridge University Press*, 1–21. <https://doi.org/10.1017/S0263574720001071>

Jiménez, E., Servín, D., Servín, R., Ochoa, F. J., Acosta, M., & Luna, G. (2021b). Modeling in two configurations of a 5R 2-DoF planar parallel mechanism and solution to the inverse kinematic modeling using artificial neural network. *IEEE ACCES*. Vol. 9, pp. 68583-68594

Kumar Saha, S. (2011). Introducción a la robótica. McGraw-Hill España. <https://elibro.net/es/ereader/ulsanoroeste/36580?page=34>

Matich, D. J. (2001, marzo). *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Universidad Tecnológica Nacional – Facultad Regional Rosario. [https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5\\_anio/orientadora1/monografias/matich-redesneuronales.pdf](https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monografias/matich-redesneuronales.pdf)

Marzal Varó, A. García Sevilla, P. y Gracia Luengo, I. (2016). Introducción a la programación con Python 3. Castelló de la Plana, Spain: D - Universitat Jaume I. Servei de Comunicació i Publicacions. Recuperado de <https://elibro.net/es/ereader/ulsanoroeste/51760?page=23>.

Meseguer González, P. y Ramon López de Mántaras Badia (2017). Inteligencia artificial. Madrid, Spain: Editorial CSIC Consejo Superior de Investigaciones Científicas. Recuperado de <https://elibro.net/es/ereader/ulsanoroeste/42319?page=127>.

Monterde, U. M. (s. f.). *Lenguajes de Programación*. Universidad Nacional Autónoma de México. Recuperado 29 de marzo de 2021, de [https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod\\_resource/content/1/contenido/index.html](https://programas.cuaed.unam.mx/repositorio/moodle/pluginfile.php/1023/mod_resource/content/1/contenido/index.html)

Nápoles Alberro, A. (2011). Análisis de mecanismos cinemática y dinámica. Delta Publicaciones. Recuperado de <https://elibro.net/es/ereader/ulsanoroeste/167035?page=32>.

Navarrete García, J. (2003, enero). *Mejora en el algoritmo de segmentación para el reconocimiento de caracteres de telegramas escritos por el Gral. Porfirio Díaz* (TFG). Universidad de las Américas Puebla. [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/navarrete\\_g\\_j/](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/navarrete_g_j/)

Ortíz González, F. R. (2008). *Apuntes Álgebra Números Complejos* (1.ª ed., Vol. 1) [Libro electrónico]. Universidad Nacional Autónoma de México. [http://ingenieria.aragon.unam.mx/fesarbook/uploads/libros/29/pdf/N%C3%BAmeros\\_Complejos.pdf](http://ingenieria.aragon.unam.mx/fesarbook/uploads/libros/29/pdf/N%C3%BAmeros_Complejos.pdf)

Ramírez, J. L., y Rubiano, A. (2012, 15 septiembre). *Modelamiento matemático de la cinemática directa e inversa de un robot manipulador de tres grados de libertad1*. *Revistas*

de investigación de la Universidad Cooperativa de Colombia.

<https://revistas.ucc.edu.co/index.php/in/article/download/47/48/>

Rivero Mendoza, F. (2001). *Una Introducción a los Números Complejos* (1.<sup>a</sup> ed., Vol. 1) [Libro electrónico]. Universidad de los Andes.

<https://www.uv.mx/personal/aherrera/files/2014/08/01a.-INTRODUCCION-A-LOS-NUMEROS-COMPLEJOS.pdf>

Rodríguez, J. (s. f.). *Modelo cinemático y control de un brazo robótico imprimible*.

Universidad Carlos III de Madrid. Recuperado 8 de enero de 2021, de [https://e-](https://e-archivo.uc3m.es/bitstream/handle/10016/16924/TFG_Juan_Carlos_Rodriguez_Zambrana.pdf?sequence=3&isAllowed=y)

[archivo.uc3m.es/bitstream/handle/10016/16924/TFG Juan Carlos Rodriguez Zambrana.](https://e-archivo.uc3m.es/bitstream/handle/10016/16924/TFG_Juan_Carlos_Rodriguez_Zambrana.pdf?sequence=3&isAllowed=y)

[pdf?sequence=3&isAllowed=y](https://e-archivo.uc3m.es/bitstream/handle/10016/16924/TFG_Juan_Carlos_Rodriguez_Zambrana.pdf?sequence=3&isAllowed=y)

Rodríguez López, J. M. (2017). *Métodos numéricos para la aproximación de raíces múltiples*

*de ecuaciones no lineales*. Universidad de Salamanca.

[https://gredos.usal.es/bitstream/handle/10366/137629/TG\\_RORIGUEZ%20LOPEZ%2C%20](https://gredos.usal.es/bitstream/handle/10366/137629/TG_RORIGUEZ%20LOPEZ%2C%20Juan%20Manuel_Metodos%20numericos.pdf?sequence=1&isAllowed=y)

[Juan%20Manuel Metodos%20numericos.pdf?sequence=1&isAllowed=y](https://gredos.usal.es/bitstream/handle/10366/137629/TG_RORIGUEZ%20LOPEZ%2C%20Juan%20Manuel_Metodos%20numericos.pdf?sequence=1&isAllowed=y)

Ruvalcaba Coyaso, F. J. (2015). *Lógica difusa para la toma de decisiones y la selección de*

*personal*. Universidad del Rosario.

<https://revistas.urosario.edu.co/xml/1872/187243745010/index.html>

Salesforce Latinoamérica. (2018, 10 abril). *¿Qué es la Cuarta Revolución Industrial?* Blog de

Salesforce. [https://www.salesforce.com/mx/blog/2018/4/Que-es-la-Cuarta-Revolucion-](https://www.salesforce.com/mx/blog/2018/4/Que-es-la-Cuarta-Revolucion-Industrial.html)

[Industrial.html](https://www.salesforce.com/mx/blog/2018/4/Que-es-la-Cuarta-Revolucion-Industrial.html)

Sánchez, R. E., González, J. J., Castillo, E., y Balmaceda, A. L. (2015). Análisis del desempeño cinetostático de un robot paralelo tipo Delta reconfigurable. *Ingeniería, investigación y tecnología*, 16(2), 2013-2224. <https://www.revistaingenieria.unam.mx/numeros/v16n2-06.php>

Seminario V., R. (s. f.). *METODOS NUMERICOS PARA INGENIERIA*. Universidad Nacional de Colombia. Recuperado 7 de enero de 2021, de <https://disi.unal.edu.co/~lctorress/MetNum/LiMetNu2.pdf>

Sepulveda, E. M. (2016, julio). *Distancia y Desplazamiento*. FisicaenLinea. <https://www.fisicaenlinea.com/04cinematica/cinematica03-distydespla.html>

*Sistema de Coordenadas Cartesianas Tridimensionais*. (2016). [Imágen]. SlideShare. <https://www.slideshare.net/debveir/aula-de-topografia-1-mdulo-1-engenharia-civil>

Texson Alarcón, G. G. (2005). *Diseño de un simulador de vuelo para la compra y venta de acciones en el mercado accionario mexicano* (TFG). Universidad de las Américas Puebla. [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lcp/texson\\_a\\_gg/](http://catarina.udlap.mx/u_dl_a/tales/documentos/lcp/texson_a_gg/)

Torres Vargas, L. Z. (2012). *Introducción a la robótica*. Grupo Editorial Éxodo. Recuperado de: <https://elibro.net/es/ereader/ulsanoroeste/128561?>

Trigo Aranda, V. (s. f.). *Algoritmos*. Autores científico-técnicos y académicos. Recuperado 4 de marzo de 2021, de <https://www.acta.es/medios/articulos/matematicas/035041.pdf>

Universitat Politècnica de Catalunya. (2012). *Fundamentos de MATLAB*. UPCommons.

[https://upcommons.upc.edu/bitstream/handle/2117/192916/tema\\_1\\_fundamentos\\_de\\_matlab-5148.pdf?sequence=1&isAllowed=y](https://upcommons.upc.edu/bitstream/handle/2117/192916/tema_1_fundamentos_de_matlab-5148.pdf?sequence=1&isAllowed=y)

Vázquez Fernández-Pacheco, A. S. Ramos de la Flor, F. y Fernández Rodríguez, R. (2015).

Robótica educativa. RA-MA Editorial. Recuperado de <https://elibro.net/es/ereader/ulsanoroeste/106572?page=42>.

Vivas Fernández, L., y Sáez López, J. M. (2019). Integración de la robótica educativa en Educación Primaria. *Revista Latinoamericana de Tecnología Educativa*, 18(1), 107–128.

<https://doi.org/10.17398/1695-288X.18.1.107>