

**DISEÑO Y CONTROL DE UN PROTOTIPO ROBOTIZADO PARA LA
REHABILITACIÓN DE LESIONES DE RODILLA ENFOCADO AL
FORTALECIMIENTO MUSCULAR**

JORGE ENRIQUE ARAQUE ISIDRO

**UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
MAESTRÍA EN CONTROLES INDUSTRIALES
PAMPLONA
2021**

**DISEÑO Y CONTROL DE UN PROTOTIPO ROBOTIZADO PARA LA
REHABILITACIÓN DE LESIONES DE RODILLA ENFOCADO AL
FORTALECIMIENTO MUSCULAR**

JORGE ENRIQUE ARAQUE ISIDRO

TESIS

PhD. CÉSAR AUGUSTO PEÑA CORTES
Docente de la Universidad de Pamplona

PhD. LEONARDO MEJÍA RINCÓN
Docente de la Universidad Federal de Santa Catarina, Florianópolis, Brasil

UNIVERSIDAD DE PAMPLONA
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
MAESTRÍA EN CONTROLES INDUSTRIALES
PAMPLONA
2021

NOTA DE ACEPTACIÓN

DEDICATORIA

A mis padres que han dedicado parte de su vida a apoyarme y educarme para que yo pueda cumplir mis sueños y metas, sin ellos nada de lo que he logrado en mi vida sería posible. Para ellos este y todos los logros que obtenga a lo largo de mi existencia.

AGRADECIMIENTOS

A la Universidad de Pamplona por permitirme formarme como Ingeniero en Mecatrónica y permitirme realizar la Maestría en Controles Industriales y brindarme las herramientas necesarias y el talento humano para mi formación.

Al profesor Cesar Augusto Peña Cortes por sus enseñanzas a lo largo de mi proceso académico en el pregrado para formarme como Ingeniero en Mecatrónica y en la maestría en Controles Industriales, especialmente por su apoyo en mi trabajo de grado de la maestría como director de tesis. Por el tiempo y conocimientos compartidos y la confianza dada en el desarrollo de este proyecto.

Al profesor Gonzalo Guillermo Moreno por su apoyo y asesoría en el desarrollo de mi tesis de maestría.

A la Universidade Federal de Santa Catarina por permitirme ser parte de su cuerpo estudiantil y brindarme las herramientas y conocimientos necesarios para completar mi trabajo de grado de maestría.

Al profesor Leonardo Mejía Rincón por darme la confianza para realizar la pasantía investigativa en la Universidade Federal de Santa Catarina en Florianópolis, Brasil en el programa de posgrado de Ingeniería Mecánica, en donde adquirí los conocimientos necesarios para completar mi tesis de maestría.

CONTENIDO

Resumen.....	13
Introducción	16
Capítulo I	19
Generalidades.....	19
1. Planteamiento del problema y Justificación	19
Problema científico:	19
Objetivo General.....	20
Objetivos Específicos.....	20
Metodología	20
Capitulo II.....	23
Marco Teórico.....	23
Aplicaciones de la robótica en el área de la salud	¡Error! Marcador no definido.
Movimiento de rodilla.....	23
La rehabilitación	23
Síntesis de mecanismos.....	26
Síntesis de Numero:	28
Conceptos básicos de síntesis de números.....	28
Procedimientos para la síntesis de número	28
Síntesis de tipo	29
Síntesis dimensional.....	29
Teoría de grafos	30
Teoría de la red de vectores	31
Capítulo III.....	41
Movimientos de flexo-extensión para la rehabilitación de lesiones de rodilla.....	41
Análisis de la trayectoria.....	41
Trayectoria directa	47
3.1. Mecanismo que imita la trayectoria en los movimientos de flexo-extensión	50
Síntesis del mecanismo:	50
Cinemática inversa.....	54
Cinemática directa	57
Cinemática inversa.....	61

Cinemática Directa.....	66
Optimización del mecanismo.....	69
Optimización por PSO (Particle Swarm Optimization).....	34
Función de vuelo.....	36
Método de Davies	36
I. Cinemática:	37
II. Estática:	39
Aplicación del método de Davies	73
Estática del mecanismo sin pierna (mecanismo de 5 barras).....	87
3.2. Control de posición de las barras del mecanismo que permita establecer arcos de movilidad y tiempos que se deben aplicar para cada uno de los movimientos necesarios en la rehabilitación de lesión de rodilla.	111
Modelamiento de Motor DC.....	111
Modelado analítico de motor DC Caso 1.....	112
Ecuación mecánica.....	114
Ecuaciones de modelo analítico.....	114
Ecuación mecánica.....	115
Modelado analítico de Motor DC Caso 2	117
Modelado analítico de Motor DC Caso 3	120
Modelado experimental de un motor DC Caso 1.....	122
Modelo no lineal de un motor DC	123
Toma de datos Motor DC	128
Aplicación de los modelos de motor DC	135
Modelo analítico de motor DC Caso 2	137
Modelo analítico del motor DC Caso 3	138
Modelo experimental del Motor DC Caso 1	140
Modelo de motor DC usando la herramienta Ident de Matlab.....	141
Análisis de Respuesta de Modelo de los Motores DC.....	142
Señal de Prueba.....	142
Análisis del Error	144
Sintonización del Regulador PID para controlar los Motores DC.....	144
Tipo de Controlador.....	145
Control Cinemático.....	146
3.3. Prototipo robotizado que permita efectuar los movimientos de flexo-extensión de acuerdo a los ejercicios planteados para la terapia de rehabilitación.	147

Construcción de la estructura mecánica del prototipo	147
Construcción de la electrónica de control del prototipo	151
CAPÍTULO IV.....	161
Resultados	161
Conclusiones	173
Bibliografía	176
ANEXOS	179

LISTA DE ILUSTRACIONES

Ilustración 1. Metodología de proyecto de mecanismo	27
Ilustración 2. Demostración helicoides.....	32
Ilustración 3. Trayectorias de tobillo y rodilla.....	41
Ilustración 4. Marcas en el paciente utilizadas para analizar la trayectoria desarrollada por la rodilla.	42
Ilustración 5. Camino desarrollado por la rodilla.	43
Ilustración 6. Curva que describe la trayectoria del tobillo	45
Ilustración 7. Análisis de trayectoria de la pierna.	45
Ilustración 8. Trayectorias calculadas.....	49
Ilustración 9. Síntesis de tipo.....	52
Ilustración 10. Síntesis de tipo.....	53
Ilustración 11. Formación de eslabones.....	53
Ilustración 12. Cinemateca inversa.....	54
Ilustración 13. Análisis geométrico mecanismo de 5 barras sin punto de interés	55
Ilustración 14. Cinemateca directa.....	57
Ilustración 15. Síntesis de tipo (error).....	60
Ilustración 16. Síntesis de Tipo (correcto).....	60
Ilustración 17. Mecanismo con dos circuitos.....	60
Ilustración 18. cinemática inversa.....	61
Ilustración 19. Análisis geométrico del mecanismo de 5 barras con punto de interés 1	62
Ilustración 20. Análisis geométrico del mecanismo de 5 barras con punto de interés 2	63
Ilustración 21. Análisis geométrico del mecanismo de 5 barras con punto de interés 3	64
Ilustración 22. Análisis geométrico del mecanismo de 5 barras con punto de interés 4	65
Ilustración 23. Análisis geométrico del mecanismo de 5 barras con punto de interés 5	66
Ilustración 24. Cinemática directa	67
Ilustración 25. Análisis geométrico del mecanismo de 5 barras con punto de interés 6	68
Ilustración 26. Optimización por enjambre de partículas	71
Ilustración 27. Representación esquemática del mecanismo de 5 barras sistema inercial 0_{XYZ} .	74
Ilustración 28. Red de acoples del mecanismo	74
Ilustración 29. Grafo de acoples GC del mecanismo.	75
Ilustración 30. Vectores de posición S_0 y los respectivos movimientos en el mecanismo.	76
Ilustración 31. Grafo de movimientos GM del mecanismo.	77
Ilustración 32. Determinación del f circuito del mecanismo.	78
Ilustración 33. Acciones en los acoples del mecanismo.	88
Ilustración 34. Grafo de acciones del mecanismo de 5 barras.	89
Ilustración 35. f cortes K_1, K_2, K_3 y K_4 en el grafo de acciones del mecanismo.....	90
Ilustración 36. Representación esquemática del mecanismo con pierna.	97
Ilustración 37. Red de acoples del mecanismo	97
Ilustración 38. Vectores de posición S_0 y los respectivos movimientos.	99
Ilustración 39. Grafo De Movimientos GM Del Mecanismo.	100
Ilustración 40. Determinación de los f circuitos del mecanismo.	101
Ilustración 41. Grafo de acciones del mecanismo	105
Ilustración 42. f cortes K_1, K_2, K_3, K_4, K_5 y K_6 en el grafo de acciones del mecanismo.....	106

Ilustración 43. Análisis Matemático del Motor DC.....	112
Ilustración 44. Modelo matemático del motor DC	113
Ilustración 45. Modelo analítico caso 1 Motor DC 1.....	116
Ilustración 46. Modelo analítico caso 1 Motor DC 2.....	117
Ilustración 47. Modelo analítico caso 2 sin perturbación de carga.....	119
Ilustración 48. Modelo analítico caso 3	121
Ilustración 49. Modelo Experimental Caso 1	122
Ilustración 50. Modelo Experimental Caso 1 Simplificado.....	123
Ilustración 51. Respuesta modelo analítico caso 1 del motor DC 1	136
Ilustración 52. Respuesta modelo analítico caso 1 del motor DC 2	136
Ilustración 53. Respuesta modelo analítico caso 2 del motor DC 1	137
Ilustración 54. Respuesta modelo analítico caso 2 del motor DC 2	137
Ilustración 55. Respuesta modelo analítico caso 3 del motor DC 1	138
Ilustración 56. Respuesta modelo analítico caso 1 del motor DC 2	138
Ilustración 57. Respuesta del modelo del motor DC 1 simplificado	139
Ilustración 58. Respuesta del modelo del motor DC 2 simplificado	139
Ilustración 59. Respuesta modelo experimental caso 1 del motor DC 1	140
Ilustración 60. Respuesta modelo experimental caso 1 del motor DC 2	140
Ilustración 61. Modelado de los motores DC Ident	141
Ilustración 62. Respuesta modelo ident Motor DC 1.....	141
Ilustración 63. Respuesta modelo ident Motor DC 2.....	142
Ilustración 64. Respuesta a escalón Motor DC 1	143
Ilustración 65. Respuesta a escalón Motor DC 2.....	144
Ilustración 66. Análisis gráfico del sistema por el primer método de Ziegler-Nichols Motor DC 1.....	145
Ilustración 67. Análisis gráfico del sistema por el primer método de Ziegler-Nichols Motor DC 2.....	146
Ilustración 68. Medida de Separación entre los ejes de los motores	148
Ilustración 69. Eslabón binario que conecta a los ejes de los motores	148
Ilustración 70. Eslabón binario que conecta al eslabón ternario.....	149
Ilustración 71. Eslabón ternario que conecta a la pierna del paciente	150
Ilustración 72. Materiales utilizados	151
Ilustración 73. Tarjeta ESP32 WIFI.....	152
Ilustración 74. Tarjeta de desarrollo Arduino Mega 2560.....	153
Ilustración 75. Drivers para motores DC Monster Moto Shield Chip VNH2SP30.....	154
Ilustración 76. Conversor de nivel lógico	155
Ilustración 77. Conversor analógico digital HX711	155
Ilustración 78. Sensor de carga Tipo S.	156
Ilustración 79. Encoder incremental de alta resolución	156
Ilustración 80. Fuente de entrada de 110 V AC y salida de 12 V DC de tres canales a 20 A .	157
Ilustración 81. Fuentes regulables DC a DC.....	158
Ilustración 82. Motores DC.....	158
Ilustración 83. Placa electrónica.	159
Ilustración 84. Logo y nombre del prototipo robótico.....	159
Ilustración 85. Prototipo terminado	160
Ilustración 86. Análisis biomecánico del paciente referencia.....	161

Ilustración 87. Análisis de los datos tomados.	161
Ilustración 88. Gráfica de las trayectorias del paciente referencia (Paciente 0) y de los pacientes ejemplo.	164
Ilustración 89. Mecanismo con pierna obtenido con	165
Ilustración 90. Prototipo del robot construido.	170
Ilustración 91. Captura del funcionamiento del robot con el paciente.	171

LISTA DE TABLAS

Tabla 1. Medidas antropométricas de pacientes	48
Tabla 2. Características de R_a , L_a y V_a de los motores DC.....	129
Tabla 3. Características de PWM , VIN , EB e I	129
Tabla 4. Constante de te	130
Tabla 5. Velocidades angulares	130
Tabla 6. Constante eléctrica y mecánica Ke y Km	131
Tabla 7. Características tm Jm y Bm	132
Tabla 8. Características modelo experimental.....	132
Tabla 9. Datos para la herramienta Ident	133
Tabla 10. Respuesta a señal de entrada función escalón de los modelos de Motores DC.....	143
Tabla 11. Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta (primer método).	145
Tabla 12. Valores de T y L de los Motores DC	146
Tabla 13. Puntos en el eje x y eje y con el ángulo de arco de movilidad del paciente usado como referencia.....	164
Tabla 14. Dimensiones obtenidas con la optimización.....	166
Tabla 15. Pares requeridos en los motores del robot para el paciente referencia.....	167

INDICE DE ANEXOS

Anexo 1. Función código trayectorias Matlab.....	179
Anexo 2. Función transformar trayectorias Matlab	180
Anexo 3. Trayectorias tobillo Matlab	181
Anexo 4. Función trayectoria directa Matlab	183
Anexo 5. Función trayectoria inversa Matlab.....	184
Anexo 6. Función cinemática directa mecanismo sin pierna Matlab	185
Anexo 7. Función cinemática inversa mecanismo sin pierna Matlab.....	187
Anexo 8. Función dibujar robot mecanismo sin pierna Matlab.....	189
Anexo 9. Función matriz de transformación homogénea rotación en Z mecanismo sin pierna Matlab	192
anexo 10. Función matriz de transformación homogénea traslación en X mecanismo sin pierna Matlab	193
anexo 11. Función matriz de transformación homogénea traslación en Y mecanismo sin pierna Matlab	194
Anexo 12. Optimización mecanismo sin pierna Matlab.....	195
Anexo 13. Función verificación mecanismo sin pierna Matlab	196
Anexo 14. Función cinemática directa mecanismo con pierna Matlab	198
Anexo 15. Función cinemática inversa mecanismo con pierna Matlab.....	200
Anexo 16. Función dibujar robot mecanismo con pierna Matlab.....	204
Anexo 17. Optimización mecanismo con pierna Matlab.....	207
Anexo 18. Función verificación mecanismo con pierna Matlab	208
Anexo 19. Método de Davies mecanismo sin pierna Matlab	210
Anexo 20. Método de Davies mecanismo con pierna Matlab	213
Anexo 21. Código programación Arduino Mega	217
Anexo 22. Código programación Arduino ESP32.....	224
Anexo 23. Código programación prueba de encoders y motores Arduino Mega.....	235
Anexo 24. Estapa 8a mecaniso con pierna.....	238
Anexo 25. Etapa 9a mecanismo con pierna	242

Resumen

Las lesiones de rodilla se presentan muy frecuentemente en deportistas de alto rendimiento debido a los grandes esfuerzos articulares a los cuales son frecuentemente sometidos, pero también se presentan en personas de distintas edades por otros factores. A pesar de las diferentes lesiones que se pueden presentar todos sufren atrofia de los grupos musculares que se encuentran cerca de la articulación y por lo tanto se debe realizar una rehabilitación de estos músculos durante el proceso de recuperación de la lesión. El proyecto tiene como objetivo general diseñar un prototipo robotizado controlado por inteligencia artificial para la rehabilitación de lesiones de rodilla enfocado al fortalecimiento muscular. El prototipo diseñado simula los diferentes movimientos ejecutados por parte del profesional de la salud durante el proceso de rehabilitación, siendo estos: contracción isométrica, movimiento pasivo, movimiento activo libre y movimiento activo resistivo, buscando recuperar la movilidad de la articulación y fortalecer la musculatura del paciente.

Para el diseño del prototipo se realizó un análisis del movimiento en pacientes llevados a cabo por un fisioterapeuta. Los procedimientos fueron grabados en vídeo, permitiendo analizar estos movimientos en un software de análisis biomecánico. De este análisis se pudo extraer datos valiosos como lo fueron las trayectorias descritas por los pacientes, las cuales fueron la base para comenzar el diseño del mecanismo. Después de analizadas estas trayectorias se realizó una síntesis de mecanismos que permitió determinar un tipo de mecanismo apto para conseguir los movimientos deseados para realizar rehabilitación. Para conseguir las dimensiones adecuadas que permitieran generar los movimientos correctos para diferentes pacientes se realizó una optimización, se hicieron unos cálculos que permitieron obtener los torques requeridos y así se pudo conocer las características deseadas en los actuadores. Después se obtuvo el modelado de los actuadores y se logró encontrar un algoritmo de control clásico PID que permitiera llevar a cabo movimientos controlados en el prototipo.

Abstract

Knee injuries occur very frequently in high-performance athletes due to the great joint efforts to which they are frequently subjected, but they also occur in people of different ages due to other factors. Despite the different injuries that can occur, all suffer atrophy of the muscle groups that are close to the joint and therefore a rehabilitation of these muscles must be carried out during the process of recovering from the injury. The general objective of the project is to design a robotic prototype controlled by artificial intelligence for the rehabilitation of knee injuries focused on muscle strengthening. The designed prototype simulates the different movements executed by the health professional during the rehabilitation process, these being: isometric contraction, passive movement, free active movement and resistive active movement, seeking to recover joint mobility and strengthen the muscles of the patient.

For the design of the prototype, an analysis of movement in patients was carried out by a physiotherapist. The procedures were recorded on video, allowing these movements to be analyzed in biomechanical analysis software. Valuable data could be extracted from this analysis, such as the trajectories described by the patients, which were the basis for starting the design of the mechanism. After analyzing these trajectories, a synthesis of mechanisms was carried out that made it possible to determine a type of mechanism suitable to achieve the desired movements for rehabilitation. To achieve the appropriate dimensions that would allow generating the correct movements for different patients, an optimization was carried out, some calculations were made that allowed obtaining the required torques and thus the desired characteristics of the actuators could be known. Afterwards, the actuators were modeled and a classic PID control algorithm was found that would allow controlled movements to be carried out in the prototype.

Introducción

La ayuda de la robótica en los trastornos del movimiento se ha investigado ampliamente en la rehabilitación de pacientes, los procedimientos de rehabilitación con características claves como la eficacia de los rendimientos y repetitividad para tareas de movimiento de rehabilitación asistida con movimientos específicos puede permitir la recuperación progresiva de funciones de movimiento específicas y lograr la recuperación de la fuerza muscular [1][2].

Las lesiones de rodilla son una de las afecciones más comunes en deportistas y personas con problemas de salud osteoarticulares, a pesar de que existen diferentes tipos de lesiones con diferentes tipos de terapias y tiempos de rehabilitación, en común tienen que en cada una de ellas se presenta atrofia muscular de los grupos musculares extensores (cuádriceps) y los grupos musculares flexores (isquiotibiales, pata de ganso, músculo poplíteo, gastrocnemio la fascia lata), los cuales hacen posible el movimiento de flexo-extensión de la rodilla [15].

El protocolo a seguir en la rehabilitación de movilidad y fuerza muscular de lesión de rodilla es el de realizar una serie de ejercicios repetitivos que gradualmente ayuden a recuperar la articulación, esto siendo medido por fuerza muscular y los arcos de movilidad que se presenten en la rodilla. Estos movimientos de flexo-extensión se pueden dividir en tres, el movimiento pasivo el cual es realizado por parte del terapeuta sin intervención alguna del paciente, el movimiento activo libre el cual realiza el paciente sin ningún tipo de carga o fuerza contraria al movimiento y el movimiento activo resistivo el cual realiza el paciente con carga o fuerza contraria al movimiento [15].

La presente investigación se realizó con el objeto de brindar soluciones y asistencia a los profesionales de la salud, los cuales se encargan de la rehabilitación de pacientes después de sufrir algún tipo de lesión, enfermedad, o otros traumas que generen desordenes de movimiento causados por atrofia muscular, en el caso específico afectando la articulación de la rodilla, ya que, al realizar las terapias de rehabilitación los fisioterapeutas por su condición humana se les hace imposible generar movimientos y fuerzas iguales en cada repetición, en este caso de flexo extensión. Como consecuencia no pueden determinar con exactitud arcos de movilidad y sobre

esfuerzos presentes en el paciente que pueden desencadenar daños más graves, que haga que su recuperación sea ineficiente.

Con este prototipo robótico se pretende que los fisioterapeutas puedan ejercer movimientos repetitivos iguales y contar con información sobre los arcos de movilidad y esfuerzos hechos por el paciente que sea más precisa pudiendo trasladar los resultados al campo investigativo en salud y como consecuencia generar avances para una rehabilitación más efectiva que lleven a menos tiempo de recuperación.

En primer lugar, se realizó un análisis biomecánico del movimiento en la rodilla en posición supino, estos ejercicios de rehabilitación fueron hechos y supervisados por profesionales de la fisioterapia, quienes tomaron a un paciente como referencia para poder realizar el análisis, el cual se realizó colocando unas marcas en las articulaciones de los miembros inferiores (cadera, rodilla y tobillo) ejecutando repeticiones de estos movimientos siendo capturados en video para posteriormente determinar la trayectoria requerida, en este caso se determinó que el punto de referencia era el tobillo puesto que el fisioterapeuta toma al paciente de esta área.

Después de encontrar la trayectoria en el paciente de referencia, se realizan los cálculos necesarios para hallar las trayectorias con pacientes de distintas medidas antropométricas de miembros inferiores.

El mecanismo diseñado fue sintetizado a través de una metodología específica que cuenta con tres fases: Síntesis de número, síntesis de tipo y síntesis dimensional, hallando dos resultados: un mecanismo sin tener en cuenta la pierna como parte del mecanismo y el segundo, en donde se cuenta la pierna como parte del mecanismo con lo que se concluyó que era el más efectivo y mecanismo que se utilizó.

Posteriormente se utilizaron algoritmos de inteligencia artificial para optimizar las dimensiones del mecanismo teniendo como base encontrar las menores dimensiones posibles, después de la optimización se realiza el análisis cinemático y estático de los dos mecanismos analizados anteriormente para así determinar velocidades angulares y pares necesarios en los motores para

que estos puedan llevar a cabo las trayectorias deseadas, para estos cálculos se utilizó el método de Davis que se basa en teoría de grafos, teoría de redes de vectores y helicoides.

Debido al uso de motores de corriente directa para mover el prototipo robótico se hace necesario proponer un control en el cual se relacione una entrada de voltaje regulada por PWM y que tenga como salida una posición angular deseada, para esto es necesario primero obtener el modelo matemático de los motores.

Posterior a obtener el modelo matemático se encuentran las constantes de un controlador clásico PID usando los métodos clásicos, que permita al motor llegar a los ángulos calculados para cada trayectoria. Después de sintonizado el controlador se ejecuta un control cinemático el cual va a permitirle al robot encontrar una función que le permita llegar de un ángulo a otro.

Para finalizar al tener las dimensiones del robot se hace el diseño en software CAD y posteriormente se construye el prototipo, en la parte electrónica se usan diferentes tarjetas electrónicas que permiten la interacción del ser humano y el robot y esta se hace utilizando el internet de las cosas (IoT) que permite por medio de cualquier navegador interactuar con una página web, enviar la información del paciente y de la etapa de rehabilitación para ser procesada y generar los movimientos en el robot.

Se espera con el producto que pueda ser usado en la Universidad de Pamplona específicamente en el programa de fisioterapia por docentes y estudiantes con el fin de mejorar su proceso formativo e investigativa permitiendo generar lazos de interdisciplinariedad para construir conocimiento, de la misma manera se ambiciona seguir trabajando en el prototipo realizando mejoras.

Capítulo I

Generalidades

1. Planteamiento del problema y Justificación

Los profesionales de la salud que se encargan de tratar la rehabilitación de lesiones de rodilla generalmente tratan a sus pacientes con una serie de ejercicios, estos se enfocan en devolver la movilidad de la articulación y la fuerza muscular en el segmento, entre otros aspectos; más precisamente al realizar movimientos de flexo-extensión [3]. Estos profesionales cuentan con pocas máquinas que les puedan ofrecer facilidades para medir los arcos de movilidad, estos son importantes a la hora de establecer cuál es el momento adecuado para realizar el cambio de movimiento pasivo a movimiento activo libre y después al movimiento activo resistido [4].

Debido a estos inconvenientes se hace difícil determinar qué tiempos y en especial qué esfuerzos son necesarios tener para que los pacientes tengan una recuperación óptima, y no se incurra en sobre esfuerzos que generen un daño al paciente o que por el contrario no tenga el efecto deseado y por lo tanto se retrase la rehabilitación, por todo esto la inclusión de máquinas y prototipos que ayuden a realizar estos movimientos repetitivos se hace importante para mejorar la intervención en la rehabilitación de rodilla [3].

En el mercado se encuentran máquinas que se encargan en su mayor medida en el movimiento pasivo continuo, los cuales son utilizados en diferentes instituciones de salud dedicadas a la rehabilitación y en centros de investigación para su desarrollo, el concepto de este tipo de dispositivos fue introducido en la década de 1970 en donde se esperaba que el objetivo principal fuera el de reducir la espasticidad [5][6].

Problema científico:

Son escasas las aplicaciones de sistemas robotizados que den solución a la rehabilitación de lesiones de rodilla enfocadas en el fortalecimiento de los grupos musculares, de estas aplicaciones muy pocas permiten un control del progreso de las etapas de rehabilitación de un paciente y los tiempos adecuados en donde se harán los cambios de movimientos.

Objetivo General

Desarrollar un prototipo robotizado para la rehabilitación de lesiones de rodilla enfocado al fortalecimiento muscular.

Objetivos Específicos

Analizar los movimientos de flexo-extensión de rodilla en donde se observen las trayectorias del movimiento de la articulación.

Seleccionar una estructura robótica o un mecanismo que pueda imitar la trayectoria que se genera en los movimientos de flexo-extensión utilizando técnicas de inteligencia artificial para optimizar el mismo.

Realizar el control de posición de las barras del mecanismo que permita establecer arcos de movilidad y tiempos que se deben aplicar para cada uno de los movimientos necesarios en la rehabilitación de lesión de rodilla.

Construir un prototipo robotizado que permita efectuar los movimientos de flexo-extensión de acuerdo a los ejercicios planteados para la terapia de rehabilitación.

Metodología

En este estudio se realizó como primera medida terapias para la rehabilitación de rodilla a un paciente por parte de un profesional de la salud en este caso un fisioterapeuta, al realizar estas terapias se tomaron videos que mostraron de forma clara cada uno de los movimientos y repeticiones hechas con el fin de capturar esta información para posteriormente analizarla.

Después de adquirir estos videos se analizaron con el software Kinovea el cual es utilizado por deportistas, entrenadores y demás profesionales relacionados con cualquier disciplina deportiva para analizar diferentes aspectos como posiciones, trayectorias velocidades, entre otras características. En este caso se utilizó la herramienta de seguimiento de trayectorias y análisis de

velocidades de esta herramienta para obtener información acerca de las terapias de rehabilitación.

Los datos que se obtuvieron del software Kinovea se pueden exportar como hojas de cálculo que dieron información de posición contra tiempo de las trayectorias que se realizaron en los ejercicios de rehabilitación con el paciente. Estos datos serán importados a un software de control numérico la cual fue la herramienta que se usó para analizar estos datos y de esta forma determinar un mecanismo y sus medidas exactas que puedan ayudar a construir un prototipo que permita la recuperación del paciente.

Se utilizó el enjambre de partículas como la técnica de inteligencia artificial de optimización para hacer la síntesis del mecanismo que se implementó para el prototipo, al realizar estos estudios se empezó a diseñar en un software CAD el prototipo que cumpla con las características resultantes de la síntesis del mecanismo.

Antes de concluir el diseño del prototipo se determinó qué actuador iba a ser usado, el cual hizo posible el movimiento del prototipo y también generó la resistencia adecuada que permite al paciente realizar esfuerzo en el caso de que sea necesario para su recuperación.

Para la caracterización, modelamiento y control de los motores se usó una metodología experimental, basada en los conceptos matemáticos propios del modelamiento de sistemas dinámicos. Se usa el algoritmo PID para el control de los actuadores que darán el movimiento al prototipo.

La metodología aplicada fue mixta, ya que, se tomaron bases teóricas que se replicarán en cuanto a los movimientos que se usan para la rehabilitación de lesión de rodilla y también llevó parte de experimentación, debido a que se deben ver diferentes opciones que puedan ofrecer la mejor respuesta a lo requerido después de hacer el análisis a pacientes con estas afectaciones.

Se tuvo en cuenta que el diseño y construcción del prototipo debe ir sustentado por la ingeniería concurrente donde aparte de generar un producto funcional sea amigable con los usuarios finales y de allí se pueda prestar el mejor servicio.

Descripción de actividades:

- I. Búsqueda bibliográfica.
- II. Tomar videos de rehabilitación al paciente.
- III. Analizar videos con el software.
- IV. Exportar los datos del software a hoja de cálculo.
- V. Importar los datos y analizarlos.
- VI. Realizar la síntesis de mecanismos usando técnicas de inteligencia artificial.
- VII. Plantear una simulación dinámica del mecanismo
- VIII. Diseño del mecanismo en un software CAD.
- IX. Calcular los pares requeridos por los actuadores.
- X. Determinación del actuador.
- XI. Determinar e implementar la técnica de control para el actuador.
- XII. Selección de los componentes electromecánicos incluyendo los actuadores, sensores, entre otros.
- XIII. Implementar el control de los actuadores (bajo nivel).
- XIV. Desarrollar el control cinemático del mecanismo para ejecutar las trayectorias (alto nivel).
- XV. Pruebas de verificación.
- XVI. Redacción de informes y artículos.

Capítulo II

Marco Teórico

Aplicaciones de la robótica en el área de la salud

La robótica es una disciplina que ha buscado formas de facilitar los trabajos hechos por el ser humano, de manera que se hagan con mayor eficiencia [7]. En el campo de la salud no es diferente ya que se han desarrollado diferentes prototipos robóticos para aplicaciones en cirugía, atención al usuario, consulta general y también en rehabilitación de diferentes lesiones [7][8][9].

En el campo de la rehabilitación se abren muchas posibilidades debido al amplio rango de lesiones y complicaciones que se pueden encontrar en diferentes pacientes con diferentes afectaciones de salud, en donde se ha despertado el interés de desarrollar prototipos de rehabilitación en compañías de desarrollo tecnológico, instituciones y universidades de varias partes del mundo [9]. Unas de esas afectaciones que se pueden encontrar son las lesiones que se dan por distintos motivos en las articulaciones; en especial se dan estas lesiones en deportistas y personas con enfermedades de tipo degenerativo; siendo estas partes del cuerpo humano unas de las más importantes pero a su vez siendo de las estructuras biomecánicas que tienden a presentar complicaciones debido movimientos inesperados y no aptos para cada una de las articulaciones ya que estas cuentan con ciertos rangos y arcos de movilidad.

Movimiento de rodilla

La rodilla es un vínculo entre la cadera y el tobillo, concilia dos imperativos contradictorios: poseer gran estabilidad y a la vez gran movilidad. Las partes importantes de la rodilla son: huesos, ligamentos, tendones y cartílagos [10].

La rehabilitación

En los años recientes se ha despertado el interés de desarrollar máquinas de rehabilitación en compañías de desarrollo tecnológico, instituciones y universidades de varias partes del mundo para rehabilitar y fortalecer completamente la parte afectada (por ejemplo: rodilla, tobillo,

cadera, etc.); recuperar la movilidad; disminuir el trabajo repetitivo de un terapeuta; incrementar el número de sesiones de terapias; reducir el tiempo de recuperación y ofrecer una mayor diversidad de terapias personalizadas con movimientos precisos y seguros [6].

En los últimos años, la investigación sobre los sistemas de rehabilitación se ha centrado principalmente en actuadores neumáticos, debido a la similitud de su funcionamiento con los músculos humanos reales [1, 2].

Al sufrir cualquier tipo de afectación en las articulaciones como pueden ser lesiones, cirugías, accidentes cerebro-vasculares entre otras dolencias en pacientes, se debe tener en cuenta que los primeros pasos en el proceso de rehabilitación es la movilización de la extremidad para poder volver a las actividades diarias normalmente o en casos graves lo mejor posible [10]. Generalmente, los pacientes con lesiones de rodilla tienen en común la atrofia muscular de los grupos musculares extensores (cuádriceps) y los grupos musculares flexores (isquiotibiales, pata de ganso, músculo poplíteo, gastrocnemio la fascia lata), músculos encargados del movimiento de flexoextensión de la rodilla y también su estabilidad [2], [10],[11].

La terapia aplicada a los pacientes sigue un protocolo en la rehabilitación básicamente con ejercicios de movilidad y fuerza muscular que gradualmente ayuden a recuperar la articulación, esto siendo medido por fuerza muscular y los arcos de movilidad que se presenten en este caso en la rodilla. Estos movimientos de flexoextensión se pueden dividir en tres [3], el movimiento pasivo el cual es realizado por parte del terapeuta sin intervención alguna del paciente[4], el movimiento activo libre el cual realiza el paciente sin ningún tipo de carga o fuerza contraria al movimiento y el movimiento activo resistivo el cual realiza el paciente con carga o fuerza contraria al movimiento [11].

El terapeuta debido a su condición humana, no puede realizar movimientos de rehabilitación que proporcionen al paciente un arco de movilidad exacto, ni tampoco que la fuerza aplicada sean las requeridas en cada etapa de rehabilitación, esto puede generar retrasos en los tiempos de recuperación o incluso puede llegar a afectar de manera negativa la articulación y los músculos que se quieren rehabilitar, por todo esto la inclusión de sistemas robóticos que ayuden a realizar estos movimientos repetitivos [12].

Muchos centros de investigación se encuentran trabajando en máquinas que se encargan en su mayor medida en el movimiento pasivo continuo, el concepto de este tipo de dispositivos fue introducido en la década de 1970 en donde se esperaba que el objetivo principal de estos dispositivos fuera el de reducir la espasticidad y fue evolucionando hasta desarrollarse sistemas robóticos que pretenden ayudar en la rehabilitación de articulaciones [4]–[6],[12], [13].

Para empezar con el diseño de mecanismos de sistemas robóticos es necesario poder tener claras las trayectorias de cada una de las articulaciones o trayectorias objetivo a las que los mecanismos deben tratar de seguir con el menor error posible[3], [6], [12]–[14], esto se puede lograr de diferentes maneras aunque lo más efectivo es el uso de herramientas informáticas como por ejemplo software especializado en la captura de datos por medio de la toma de videos que son analizados por estos programas que generalmente se usan en deportistas para conocer aspectos de su desempeño físico [15].

Cuando se tiene la trayectoria de cada una de las articulaciones o la articulación sobre la que se va a trabajar, es necesario empezar a observar mecanismos que puedan cumplir con el objetivo de trazar una trayectoria parecida a la encontrada, teniendo en cuenta que la trayectoria debe tomarse en varios pacientes para observar un patrón similar entro todos para llegar a un mecanismo que pueda suplir las necesidades de cualquier persona que vaya a usarlo, requiriendo esto de herramientas informáticas que puedan dar aproximaciones de los datos obtenidos del software que calcula las trayectorias [15].

Los robots en rehabilitación se han introducido en las primeras fases de recuperación después del accidente cerebrovascular para superar las principales limitaciones de la terapia manual tradicional. Los robots pueden administrar terapias de rehabilitación de alta calidad con buenas

mediciones cuantitativas y cualitativas. A lo largo de los años, se han desarrollado varios dispositivos robóticos de rehabilitación de la marcha basados en diferentes conceptos [6,7]. Sin embargo, la mayoría de los sistemas robóticos de entrenamiento de la marcha, como Lokomat, ReoAmbulator [15] y el sistema de la Universidad de Auckland, integran cintas de correr y plataformas fijas [12]. Están destinados a pacientes agudos en grandes centros de rehabilitación u hospitales; y el progreso de la rehabilitación es inferior al entrenamiento de la marcha sobre el suelo.

Por lo tanto, existe una gran necesidad de sistemas robóticos portátiles para pacientes con accidente cerebrovascular crónico en centros de rehabilitación comunitarios o entornos domésticos. Estos dispositivos portátiles deben ser livianos, seguros y fáciles de poner y quitar. Varios dispositivos de rehabilitación de la marcha dirigidos específicamente a la articulación del tobillo o la articulación de la rodilla han sido desarrollado [8].

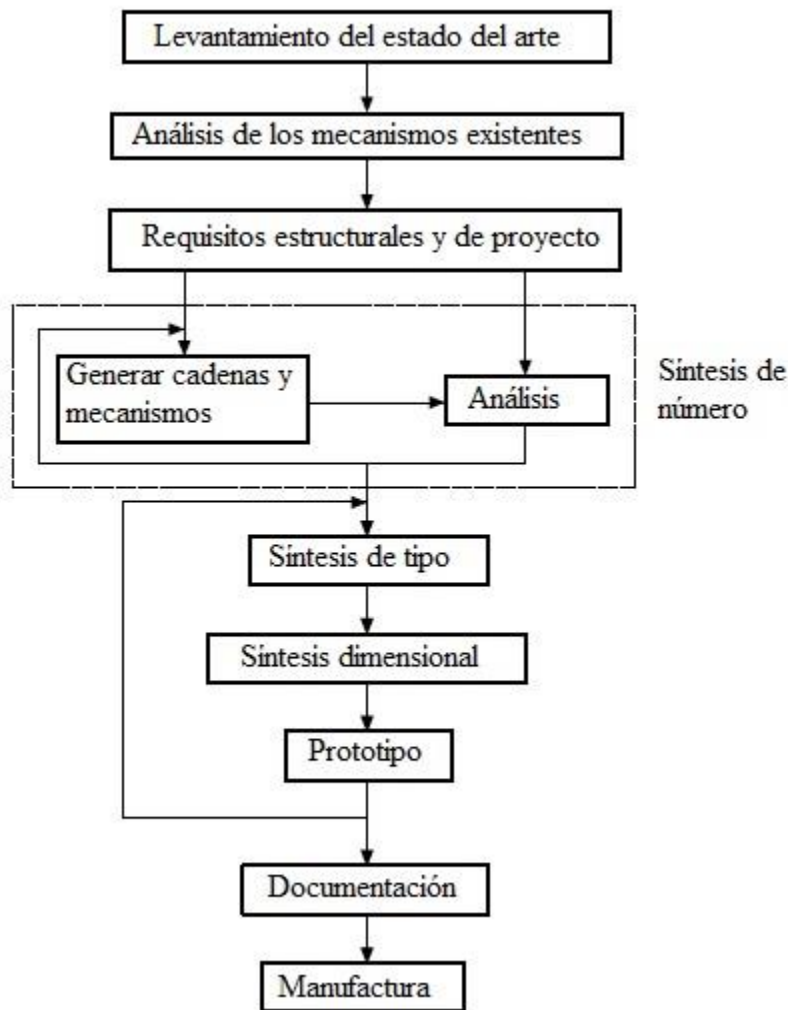
Sin embargo, el peso sigue siendo el principal desafío para lograr la portabilidad. El robot de tobillo desarrollado en el MIT [21] pesa 3,1 kg solo para la articulación del tobillo. El dispositivo de asistencia para la rodilla comercializado desarrollado por Tibion pesa más de 4,5 kg solo para la articulación de la rodilla. Una órtesis de control de la postura cuasi-pasiva compatible (CSCO) para la asistencia de la articulación de la rodilla desarrollada recientemente por Shamaei et al. pesa 3 kg . Además, los dispositivos portátiles destinados a proporcionar una torsión de asistencia activa a las articulaciones de la rodilla y el tobillo son muy raros debido al diseño de actuación ineficaz y voluminoso. La ortesis liviana de rodilla, tobillo y pie (KAFO) se intentó recientemente en la Universidad de Michigan, pero está diseñada con actuadores neumáticos atados a un compresor estacionario, por lo que el sistema no puede ser verdaderamente portátil y usarse en casa [8].

Síntesis de mecanismos

Las metodologías de proyecto de mecanismos se organizan por etapas de forma que se sistematice un proceso, hay diversas metodologías de un proceso de mecanismos, cada una de ellas presentan sus características diferenciales pero en general las estructuras son similares, la

metodología que será explicada abarca las etapas más generales, sin embargo las metodologías de proyectos de mecanismos son más específicas, por lo tanto sus etapas son vistas con más detalle [12].

Ilustración 1. Metodología de proyecto de mecanismo



Autor: [23]

Se inicia el procedimiento realizando el estado del arte, el objetivo de esta etapa es enumerar los dispositivos existentes que poseen una función similar deseada. Otro objetivo es entender mejor el problema que se desea resolver, observando qué características se tornan mejores para dar soluciones [22].

Síntesis de Numero:

Parte de un conjunto de requisitos estructurales para determinar la cantidad de eslabones, pares cinemáticos y cuáles eslabones son conectados entre sí, obteniendo diversos mecanismos que puedan satisfacer los requisitos estructurales, teniendo en cuenta los conceptos de movilidad e isomorfismo.

Conceptos básicos de síntesis de número

Algunos conceptos básicos son necesarios para comprender la síntesis de números, siendo ellos:

Partición: Es el conjunto de todos los eslabones que serán utilizados para montar una cadena cinemática, de esta forma una partición puede tener eslabones unitarios, binarios, ternarios y en adelante. En las síntesis de mecanismos, los elementos de par cinemático deben ser distribuidos entre los eslabones. De esta forma, si un eslabón recibe tres elementos de par cinemático, él será un eslabón ternario, si recibe cuatro elementos será cuaternario [22].

Variación: Es el modo en el cual los eslabones de una partición pueden ser conectados para generar una cadena cinemática, así dada una partición, al conectar los eslabones de todos los modos posibles, se generan todas las variaciones posibles.

Inversión: Es el proceso de asociar un referencial fijo a uno de los eslabones de la cadena cinemática, así partiendo de una cadena cinemática determinar todas las inversiones de la misma, sería equivalente a determinar todos los mecanismos que pueden ser generados por la cadena.

Procedimientos para la síntesis de número

La síntesis de número parte de un conjunto de requisitos estructurales, usualmente son usados la movilidad (M), la dimensión del espacio de trabajo (λ) y la cantidad de circuitos independientes (v). Con estos tres requisitos estructurales, se puede utilizar la ecuación de movilidad y la

ecuación de Euler para determinar la cantidad de eslabones (n) y la cantidad de pares cinemáticos con conectividad de par =1 (j). Con j se puede calcular la cantidad de elementos de pares cinemáticos [22].

Calcular la cantidad de eslabones y elementos de par cinemático, determina todas las particiones posibles. Teniendo todas las particiones posibles se determinan todas las variaciones (Cadenas cinemáticas) posibles. Determinadas todas las cadenas cinemáticas se puede obtener todas las inversiones (mecanismos) posibles [22].

Síntesis de tipo

Determina el tipo de cada par cinemático de un mecanismo. Inicialmente se debe enumerar todos los tipos de pares cinemáticos que pueden ser enumerados. Los factores que inviabilizan el uso de algunos tipos de pares cinemáticos son:

1. Espacio de trabajo: Los pares cinemáticos deben satisfacer las restricciones del espacio de trabajo, por ejemplo, si el espacio de trabajo fuera planar podrán ser utilizados pares inferiores de revolución, prismático y planar [23]. No se debería usar un par cilíndrico debido a que este permite movimientos que no están en el plano.
2. Fabricación: Procesos disponibles para la fabricación pueden afectar la selección de pares disponibles o viables económicamente.
3. Mantenimiento: El mecanismo debe ser de fácil mantenimiento y en caso de falla, los pares deben ser de fácil sustitución [22].

Síntesis dimensional

La síntesis dimensional puede ser clasificada en tres tipos:

- a. Generación de función. El objetivo de esta síntesis es determinar las dimensiones del mecanismo para que su comportamiento describa una función dada, de esta forma para

un valor de ángulo de entrada, la barra de salida deberá retornar un valor deseado para el ángulo de salida, como una función matemática.

- b. Generación de trayectoria. El objetivo de esta síntesis es determinar las dimensiones del mecanismo que un punto de interés recorra una trayectoria específica al accionar el mecanismo. En este tipo de síntesis se especifica la trayectoria que el punto deberá recorrer y no se especifica la orientación del eslabón a lo largo de la trayectoria.
- c. Generación de movimiento: El objetivo de esta síntesis es determinar las dimensiones de un mecanismo dado un conjunto de posiciones homologadas de un plan. En este caso se controla la posición y la orientación en apenas una cantidad finita de posiciones. Este tipo de síntesis es utilizado cuando se desea que el mecanismo pase por las posiciones determinadas con una orientación determinada, en este caso no es necesario que recorra una trayectoria [22].

Teoría de grafos

Es una herramienta muy utilizada para representar sistemas de ingeniería por dos razones básicas: Cada elemento del grafo es definido de forma que tenga correspondencia directa con los elementos del sistema, los teoremas y las rutinas aplicadas a los grafos permiten representar el comportamiento de propiedades del sistema como deformaciones, fuerzas, velocidades, movimientos entre otros [23].

Es común encontrar representaciones a través de grafos para modelar problemas de lógica, logística, estructuras, circuitos eléctricos, sistemas mecánicos entre otros. Por tanto, se encuentran diagramas compuestos de aristas y vértices asociados a valores y/o nomenclaturas. En general son utilizados dos abordajes: grafos lineales y grafos de conexión, para representar problemas que envuelven sistemas físicos concretos en el espacio. Los grafos de conexión modelan el flujo de energía que envuelve un problema, entre tanto los grafos lineales permiten modelar problemas en diferentes dominios que no necesariamente son de energía [23].

A pesar de que los grafos lineales son más fácilmente encontrados en la literatura ambos abordajes tienen esencialmente capacidades idénticas como herramientas de modelaje y las diferencias existentes son superficiales [23].

Teoría de la red de vectores

Es una extensión de las aplicaciones de la teoría de grafos para sistemas eléctricos, empleado para solucionar la dinámica de sistemas mecánicos discretos en el espacio tridimensional. Por tanto combina la teoría de grafos y el cálculo vectorial, donde cada elemento físico es representado por un vector de desacople y análogamente en una arista en el grafo [23].

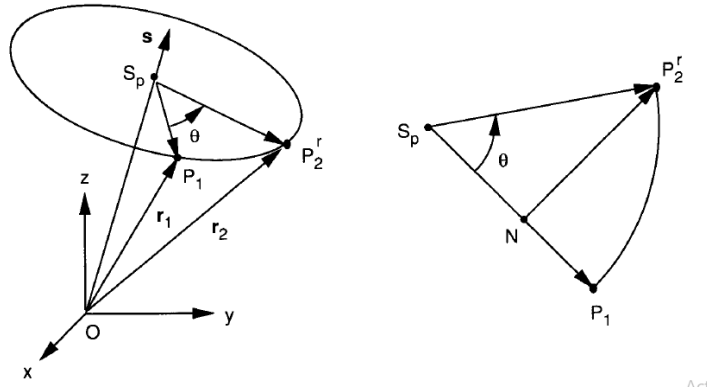
El procedimiento de la teoría de red de vectores es basado en las leyes de Kirchhoff de corriente y tensión conteniendo dos tipos de ecuaciones: Topológicas que describen la interconexión entre los cuerpos en el sistema y terminales que describen las características físicas de los cuerpos del sistema [23].

La aplicación de esta en un sistema mecánico es hecha a partir de un diagrama de red de vectores donde cada vértice corresponde a un punto de interconexión entre los cuerpos y cada arista corresponde a un vector de desacople que representa algún elemento físico. Esta teoría es una formulación rama-cuerda usada para derivar sistemáticamente las ecuaciones de movimiento en términos de coordenadas absolutas para sistemas tridimensionales de partículas de masa libre [23].

Teoría de tornillo (helicoides)

La teoría de tornillos o helicoides como se nombrarán de ahora en adelante, es muy importante para el análisis de movimientos y esfuerzos a realizar en el desarrollo del prototipo propuesto. Por este motivo se hace la demostración de esta teoría como se muestra en la ilustración 2

Ilustración 2. Demostración helicoides



Los vectores de la ilustración 2 se representan a continuación:

$$P_1 = \overline{OP_1}; \overline{OP_2}; S (S_x, S_y, S_z)$$

Operando los vectores podemos obtener.

$$\overline{S_P P_2} = r_1 - (r_1^T S) S$$

$$\overline{S_P P_2^r} = r_2 - (r_2^T S) S$$

$$\overline{NP_2^r} \perp \overline{S_P P_1}$$

Los dos vectores están en el mismo plano

$$|S_P P_1| = |S_P P_2^r| \text{ y } S \text{ x } |S_P P_1| = S \text{ x } r_1$$

$$|S_P N| = \overline{S_P P_1} (\theta \overline{NP_2^r} = S \text{ x } r_1 S \theta$$

$$\overline{NP_2^r} = \overline{S_P P_2^r} - \overline{S_P N} \Rightarrow \overline{S_P P_2^r} - \overline{S_P N} + \overline{NP_2^r}$$

Reemplazando

$$r_2 - (r_2^T S) S = [r_1 - (r_1^T S) S] (\theta + S \text{ x } r_1 S \theta$$

$$r_1^T S = r_2^T S$$

Se obtiene.

$$r_2 = r_1 (\theta + Sx r_1 S\theta + S (r_1^T S) (1 - (\theta)))$$

Se puede decir que los vectores pueden ser representados por.

$$r_1 = P_1 - S_o$$

$$r_2 = P_2 - S_o - ts \quad \overline{S_o P_2} - \bar{t}s = r_2$$

Por lo tanto el vector P_2 equivale a:

$$P_2 = S_o + ts + (P_1 - S_o) (\theta + Sx(P_1 - S_o) S\theta + [(P_1 - S_o)^T S] S (1 - (\theta))) \quad [23]$$

Optimización del mecanismo

Algunas herramientas que podemos encontrar son las técnicas de inteligencia artificial que son de gran ayuda para cumplir con este objetivo, en especial las técnicas que sirven para optimizar en este caso las dimensiones de los mecanismos escogidos como posibles opciones. Algunas de estas técnicas son los algoritmos genéticos y optimización por enjambre de partículas que son muy conocidas y ampliamente trabajadas debido a su funcionabilidad, estas técnicas de inteligencia artificial se basan a comportamientos de la naturaleza como la evolución de los individuos y los grupos de aves que se coordinan para seguir un líder, respectivamente [15][16].

Tanto en la aplicación de los algoritmos genéticos como la optimización por enjambre de partículas en la síntesis del mecanismo, se definen funciones fitness que calculan el error entre las trayectorias encontradas en el análisis de datos. Este error corresponde con la sumatoria de las distancias entre los puntos desarrollados en la trayectoria objetivo y en la trayectoria del mecanismo para un mismo vector de n posiciones del eslabón que se halla diseñado para la articulación deseada. Cuando esta función fitness se lleguen a minimizar, se podrán encontrar las dimensiones óptimas del mecanismo. Al implementar las técnicas de inteligencia artificial se deben tener en cuenta los valores mínimos y máximos de las medidas de los mecanismos que se pusieron a consideración bajo unos límites establecidos [17],[18],[19].

El uso de robots para la asistencia de terapias físicas ha ido en aumento, aunque ya se ha trabajado en este campo desde décadas atrás en los últimos tiempos el auge y avance de sistemas de control que se aplican a estos prototipos autónomos los ha hecho mucho más eficientes aportando así mejores resultados, principalmente se aplican técnicas de control clásica que sirven para dar un control de actuadores que claramente dejan en evidencia que su uso mejora el rendimiento de las terapias hechas por fisioterapeutas, aunque, todavía se generen limitaciones. En este sentido se puede decir que la aplicación de técnicas de control inteligente han venido ayudando a superar estas limitaciones y de esa forma dar aún mejores resultados que se evidencian en la mejora de pacientes [20],[3],[1].

Optimización por PSO (Particle Swarm Optimization)

La metaheurística de optimización por enjambre de partículas o por sus siglas en inglés PSO (Particle Swarm Optimization), es una técnica de optimización matemática inspirada en el comportamiento social de los individuos dentro de un grupo analizando las interrelaciones de los individuos con los integrantes del mismo [24].

Esta técnica se basa en el comportamiento que describen las coreografías observadas de las bandadas de aves y en sus reglas de vuelos sincronizados en donde se pueden observar acciones

de dispersión y de reagrupación, intentando de esta forma replicar la dinámica de grupo por medio de algoritmos basado en población [24].

Así se toma un individuo y dependiendo de su comportamiento se direcciona un grupo y se busca la optimización en este sentido. Cada individuo puede modificar su comportamiento basado en tres factores: conocimiento sobre el entorno, conocimiento histórico y experiencia de los individuos cercanos. A su vez cada individuo se encuentra influenciado por dos factores: conocimiento y habilidades propias del individuo y por el individuo líder del grupo [24].

El líder del grupo tiene características o habilidades superiores, por tanto los miembros del grupo confían en él pero este puede cambiar siempre y cuando otro individuo tenga mejores características, este individuo líder guía a los demás individuos tomando decisiones basadas en su experiencia, pero a su vez los demás miembros pueden inferir en las decisiones [18].

El PSO explora subespacios relativamente cercanos entre sí, cada individuo posee los siguientes operadores:

$X_i(t)$ Es la posición.

$V_i(t)$ Es la velocidad.

P_{best} Es la mejor posición de la partícula. (Representa la memoria autobiográfica de las partículas).

G_{best} Es la mejor posición del líder. (Tendencia de los individuos por regresar donde tuvieron la mejor respuesta en el pasado).

Si $Pbest$ es grande puede producir un incremento de individuos errantes y aislados en el espacio, si $Gbest$ es grande puede producir cúmulos atraídos prematuramente por subóptimas.

Principalmente este método fue propuesto para programación continua sin embargo existen versiones del PSO para programación discreta.

Función de vuelo

$V_i(t)$ Es la dirección de búsqueda para la partícula i

$$V(t + 1) = W V_i(t) + C1 \text{rand} (Pbest - X_i(t)) + C2 \text{rand} (Gbest - X_i(t)) \quad \text{Ec.4.73.}$$

W Es el factor de inercia.

El factor de inercia se usa para dar balance a la función debido a que cuando se presenta un factor de inercia grande, las partículas tienden a diversificarse y por consiguiente tienden a separarse más entre ellas para buscar posibles soluciones a lo que se podrían llamar partículas exploradoras y cuando el factor de inercia es pequeño las partículas tienden a intensificarse lo que significa que tienden a quedarse juntas cuando hay una posible solución y por lo cual se podrían llamar partículas colonizadoras [16], [19].

Método de Davies

En este método se tienen en cuenta los conceptos de grados de libertad, restricciones internalización de acciones externas, también se utilizan analogías de las leyes de Kirchhoff para construir la relación de dependencias entre las incógnitas como son las características mecánicas

tales como las fuerzas, velocidades, entre otras son incluidas a través de la representación por helicoides [23].

Empleando la teoría de grafos para el estudio de la cinemática y de la estática de mecanismos de forma desacoplada y basándose en el mismo modelo de grafo del mecanismo, cada uno de estos como dos análisis diferentes, estos pueden ser realizados de forma independiente.

En este método los efectos dados por la fuerza gravitacional, aceleraciones, inercias, entre otros, no son tenidos en cuenta tratándose de la cinética instantánea y de la estática de mecanismos formados por cuerpos rígidos.

En resumen el método de Davies puede ser enumerado en 9 etapas tanto para la cinemática como para la estática facilitando su aplicación para cualquier mecanismo.

I. Cinemática:

1. Caracterización del mecanismo

- a. Representación esquemática con sistema inercial de coordenadas O_{XYZ}
- b. Representación en redes de acoples (topología)

2. Caracterización de los acoples

- a. Levantamiento de las características geométricas $(\vec{S}^M, \vec{S}_o, h^M)$ y del movimiento (fi) donde \vec{S}^M es el vector de las direcciones unitarias, \vec{S}_o es el vector posición de cualquier punto del eje del helicoides relativo al origen del sistema de coordenadas, h^M paso del helicoides.

3. Topología – circuitos

- a. Representación del grafo de los movimientos G_M
- b. Determinación de la matriz de f circuitos $[B_M]_{IXF}$

4. Geometría – Heligios

- a. Construcción de los Heligios = $\M
- b. Determinación de la matriz de los movimientos = $[M_D]_{\lambda \times F}$

5. Sistema de ecuaciones

- a. Determinación de la matriz de movimientos unitarios en red = $[\widehat{MN}]_{\lambda \times F}$
- b. Ley de los circuitos $[\widehat{MN}]_{\lambda \times F} \{\vec{\Phi}\}_{F \times 1} = \{\vec{0}\}_{\lambda \times 1}$

6. Súper restricciones (Redundancias)

- a. Determinación de las C_N ecuaciones redundantes.
- b. Eliminación de las ecuaciones redundantes del sistema.

7. Separación de variables

- a. Selección de las F_N variables primarias en el vector $\{\vec{\Phi}\}_{F \times 1}$
- b. Separación de las variables primarias y secundarias en el sistema de ecuaciones.

8. Solución

- a. Calcular $[MNS]_{m \times m}^{-1}$, atribuir valores a $\{\vec{\Phi}_P\}_{F_N \times 1}$ y la obtención de la solución $\{\vec{\Phi}_S\}_{m \times 1}$.

9. Estado instantáneo de movimientos.

- a. Aplicar las magnitudes $\{\vec{\phi}\}$ a los heligros $\M de cada acople.

II. Estática:

Partiendo de un modelo físico del mecanismo el análisis estático puede ser realizado conforme a las etapas descritas.

1. Caracterización del mecanismo

- a. Representación esquemática con sistema inercial de coordenadas O_{XYZ}
- b. Representación en red de acoples (topología)

2. Caracterización de los acoples

- a. Levantamiento de las características geométricas $(\vec{S}^A, \vec{S}_0, h^A)$ y de acción (C_p, C_a) donde \vec{S}^A es el vector de las direcciones unitarias, h^A es el paso del helicoides.

3. Topología – Cortes

- a. Representación del grafo de las acciones: GA
- b. Determinación de la matriz de f cortes $[Q_A] k \times c$

4. Geometría – Helifuerzas

- a. Construcción de los Helifuerzas = $\A
- b. Determinación de la matriz de las acciones = $[\hat{A}N] \lambda x C$

5. Sistema de ecuaciones

- a. Determinación de la matriz de acciones unitarios en red = $[\hat{A}N] \lambda K x C$
- b. Ley de los cortes = $[\hat{A}N] \lambda K x C \{\bar{\Psi}\} Cx1 = \{\bar{0}\} \lambda K x 1$

6. Subrestricción (grado de libertad extra)

- a. Determinación de las F_N ecuaciones dependientes.
- b. Eliminación de las ecuaciones dependientes del sistema.

7. Separación de variables

- a. Selección de la C_N variables primarias en el vector $\{\bar{\Psi}\} Cx1$
- b. Separación de las variables primarias y secundarias en el sistema de ecuaciones.

8. Solución

- a. Calcular la matriz inversa de $[A_{NS}]_{axa}$ atribuir valores a $\{\bar{\Psi}_P\} CN x 1$ y obtención de la solución $\{\bar{\Psi}_S\} a x 1$

9. Estado instantáneo de acciones

- a. Aplicar las magnitudes $\{\bar{\Psi}\}$ a las helifuerzas $\A de cada acople.

Capítulo III

Movimientos de flexo-extensión para la rehabilitación de lesiones de rodilla

Análisis de la trayectoria

Se hizo el estudio de los movimientos de rehabilitación realizados por un fisioterapeuta a un paciente con lesión de rodilla por medio del software Kinovea [15], con la ayuda de un software matemático se pueden generar las trayectorias del tobillo y la rodilla con el fin de generar el mecanismo de rehabilitación (ver Ilustración 3).

Ilustración 3. Trayectorias de tobillo y rodilla



Tomando en consideración las trayectorias de las dos articulaciones, se encontró que el movimiento del tobillo es determinante para la trayectoria de la rodilla, al analizar los datos de la trayectoria del tobillo como se aprecia en la ilustración 3, hay diferencias considerables en cada una de las repeticiones, esto se debe al error del fisioterapeuta, dado que no cuentan con herramientas que les permitan estimar posiciones de los miembros en tiempo real, por lo tanto deben hacer esto solo con una retroalimentación visual. Esto repercute directamente en la recuperación del paciente y mientras sea realizado el movimiento por una persona difícilmente

podrá ser el mismo en cada repetición. Esto genera la necesidad de crear un instrumento que permita realizar los movimientos de manera uniforme y que controle de forma efectiva los máximos arcos de movimiento, haciendo que la rehabilitación sea más efectiva [15].

Para analizar la trayectoria de las dos articulaciones se utilizó el software Kinovea, en primera estancia se tomaron algunos videos en donde un fisioterapeuta realizaba los movimientos de flexo-extensión a un paciente con el fin de extraer datos que pudiesen servir en el análisis de trayectorias, de estos se tomó un video que contenía la mejor información visual para que el estudio de los movimientos fuera el más adecuado. En este video el paciente tenía tres marcas de color amarillo y de forma triangular (ver Ilust 3), las cuales estaban distribuidas en las zonas biomecánicas más relevantes de las extremidades inferiores las cuales son la cadera, la rodilla y el tobillo [15].

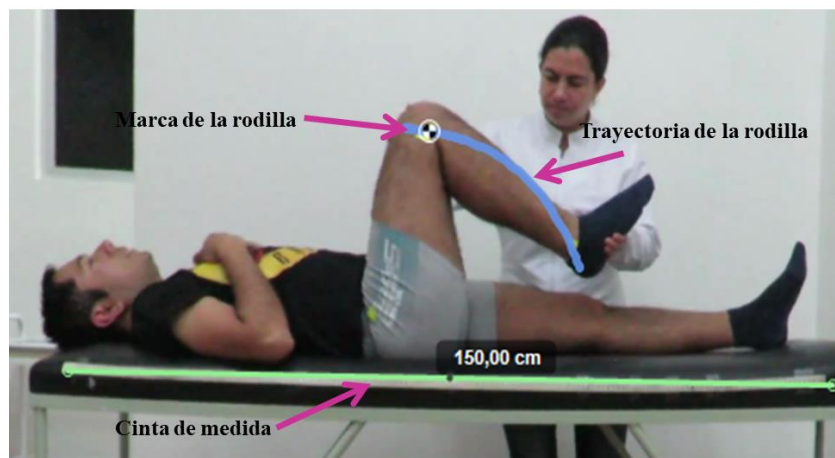
Ilustración 4. Marcas en el paciente utilizadas para analizar la trayectoria desarrollada por la rodilla.



Lo primero que se realizó es el ajuste del video a las unidades de longitud (centímetros) para lograr obtener datos que se puedan exportar y de esta forma graficar para su identificación. Esto se realizó por una cinta métrica puesta a un costado de la camilla donde se encuentra el paciente, la posición de la misma no es relevante como tampoco lo es la longitud de la cinta métrica que en este caso es de 150 cm, mediante una herramienta de la plataforma denominada *Línea* se acotó la cinta y se le da el valor de la misma en el software. Además de esto se debe colocar un punto que dé el origen de coordenadas, que en este caso será la marca del tobillo en el paciente,

se realizaron aproximadamente 10 flexoextensiones los cuales arrojaron los resultados del movimiento de la rodilla como se muestra en la ilustración 5.

Ilustración 5. Camino desarrollado por la rodilla.



La trayectoria planteada en la rehabilitación correspondiente a la rodilla se encuentra descrita en [15]. Sin embargo se puede apreciar que el arco de movimiento desarrollado por la rodilla, depende tanto de la orientación del fémur como de la orientación del grupo tibia-peroné. Por lo tanto, si se considera que la pelvis del paciente esta en reposo, el arco de movimiento de la rodilla se puede calcular plenamente al tener la localización de la rodilla y el tobillo (aplicando el teorema del coseno a estos tres puntos). Teniendo en cuenta esta consideraciones se procede a estimar la trayectoria del tobillo durante el movimiento.

Para tomar los datos de la trayectoria del tobillo se realizaron varias repeticiones en donde se observó que debido al error humano muy difícilmente se podía replicar con exactitud el movimiento que busca rehabilitar al paciente, teniendo en cuenta que esto está directamente relacionado con el arco de movilidad de la articulación haciendo que puede existir un efecto negativo en tiempos de la terapia. Estos datos se exportan dado un espacio de coordenadas euclidiano lo que hace posible graficar los puntos de la trayectoria tomada, debido al error humano de la terapia de rehabilitación se genera en realidad varias trayectorias (una para cada

repetición del movimiento) por lo cual se vio la necesidad de seleccionar la trayectoria central o promedio como la trayectoria objetivo. En la ilustración 5 se aprecia esta trayectoria.

Para obtener la trayectoria que se va a utilizar para empezar el análisis mecánico, se deben hacer los ejercicios de flexo-extensión necesarios para la rehabilitación de rodilla tanto en el arco de movilidad como en la fuerza muscular.

Para esto se busca un fisioterapeuta que aplique estos movimientos a un paciente al cual se le toman medidas antropométricas, como la estatura, peso y edad, también se deben tomar específicamente las medidas de la cadera hasta la rodilla y de la rodilla hasta el tobillo, medidas que son necesarias para realizar los cálculos correspondientes para hallar las trayectorias de personas con diferentes medidas [15].

Para poder capturar esta trayectoria se ponen marcas en la cadera, rodilla y tobillo y se hizo un video de varias repeticiones del movimiento realizado por el fisioterapeuta al paciente y este video fue analizado por el software Kinovea logrando poner medidas dentro del video para obtener los puntos dentro de las coordenadas x, y , estos puntos se extraen en un archivo excel [15].

Se planteó una caracterización de la curva descrita, se incluye el tipo y se muestran finalmente los coeficientes ilustrando el grado de exactitud o el error.

La ecuación que describe la trayectoria es la siguiente:

$$y = \frac{(P_1 * x^2) + (P_2 * x) + P_3}{x^2 + Q_1 * x + Q_2} \quad \text{Ec.3.1.}$$

En donde los coeficientes se dan debido a la caracterización de la trayectoria obtenida. Los valores de las constantes son:

$$P_1 = -793,4 \quad \text{Ec.3.2.}$$

$$P_2 = -3,047 * 10^6 \quad \text{Ec.3.3.}$$

$$P_3 = 1,36 * 10^4 \quad \text{Ec.3.4.}$$

$$Q_1 = -1,981 * 10^4 \quad \text{Ec.3.5.}$$

$$Q_2 = 6,556 * 10^4 \quad \text{Ec.3.6.}$$

Esta ecuación describe la trayectoria de un paciente que tiene como medidas 36cm de cadera a rodilla y 36cm de rodilla a tobillo, en donde a x se le dan valores de 0 a -50.

Ilustración 6. Curva que describe la trayectoria del tobillo

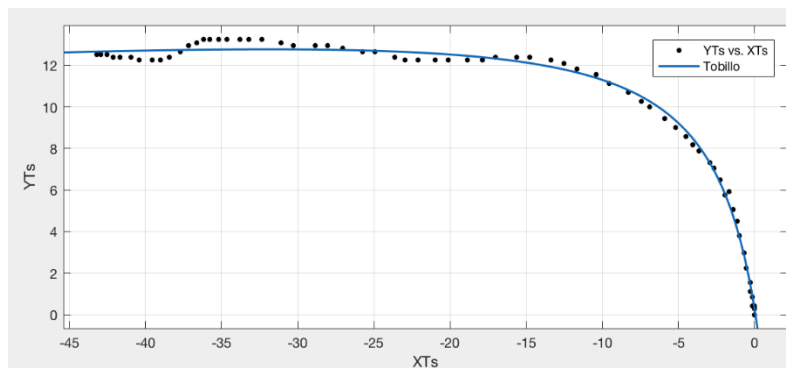
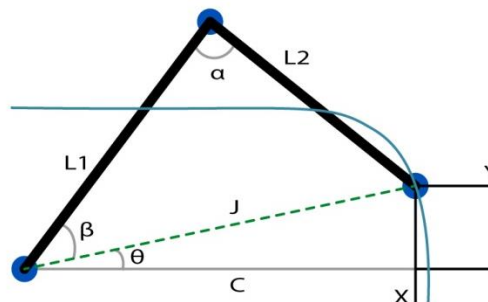


Ilustración 7. Análisis de trayectoria de la pierna.



Se hace una tarea difícil tener que hallar las trayectorias de cada paciente que va a realizar la rehabilitación, de la forma en que se hizo este estudio, por lo tanto, se calcula la trayectoria de

cada paciente teniendo en cuenta sus medidas antropométricas y teniendo en cuenta el arco de movilidad que se debe presentar al tener un desplazamiento en x, y .

Trayectoria inversa

Se calculan los ángulos presentados en el esquema cinemático (ilustración 7) con el fin de hallar el arco de movilidad que corresponde con cada una de las posiciones dadas en x, y . Así como cada ángulo del esquema θ, α y β . Esto basado en el paciente medido para conseguir los valores necesarios con el fin de calcular la posición en x, y de los pacientes con sus diferentes medidas.

Donde: x, y son conocidas.

L_1 = Es la medida de la cadera a la rodilla.

L_2 = Es la medida de la rodilla al tobillo.

Se halla la distancia C en x que hay entre los dos puntos que representan la cadera y el tobillo respectivamente.

$$C = L_1 + L_2 - x \quad \text{Ec.3.7.}$$

Cuando $x = 0$ $\beta = 180^\circ$; $\alpha = 0$; $\theta = 0$

Usando el teorema de Pitágoras se calcula θ

$$J = \sqrt{C^2 + y^2} \quad \sin \theta = \frac{y}{J} \quad \cos \theta = \frac{C}{J}$$

$$\theta = \text{atan } \theta = \text{atan } 2 \left(\frac{y}{J}, \frac{C}{J} \right) \quad \text{Ec.3.8.}$$

Usando la ley del coseno se halla α y la variable C_d indica hacia a donde es la abertura del ángulo.

$$L_2^2 = L_1^2 + J^2 - 2 * L_1 * J * \cos(\alpha) \quad C_d = 1 \quad \text{Ec.3.9.}$$

$$\cos(\alpha) = \frac{L_2^2 + J^2 - L_1^2}{2 * L_1 * J} \quad \text{Ec.10.}$$

$$\sin(\alpha) = C_d * \sqrt{1 - \cos^2 \alpha} \quad \text{Ec.3.11.}$$

$$\alpha = \text{atan2} (\sin(\alpha), \cos(\alpha)) \quad \text{Ec.3.12.}$$

Usando la ley del coseno se halla β y la variable C_d indica hacia a donde es la abertura del ángulo.

$$J^2 = L_1^2 + L_2^2 - 2 * L_1 * L_2 * \cos(\beta) \quad \text{Ec.3.13.}$$

$$\cos(\beta) = \frac{L_1^2 + L_2^2 - J^2}{2 * L_1 * L_2} \quad \text{Ec.3.14.}$$

$$\sin(\beta) = C_d * \sqrt{1 - \cos^2 \beta} \quad \text{Ec.3.15.}$$

$$\beta = \text{atan2} (\sin(\beta), \cos(\beta)) \quad \text{Ec.3.16.}$$

Donde β puede ser asociado al ángulo de arco de movilidad que se necesita medir en el paciente de la siguiente manera:

$$\text{Arco de movilidad} = 180 - \beta$$

Dando así el ángulo de la rodilla que se está recuperando.

Trayectoria directa

Se hallan los valores de x, y para determinar la trayectoria de cada paciente teniendo en cuenta sus medidas antropométricas de la cadera a la rodilla y de la rodilla al tobillo.

Donde

θ , α y β son conocidos.

L_1 = Es la medida de la cadera a la rodilla.

L_2 = Es la medida de la rodilla al tobillo.

Usando la ley del coseno se hallan x y y .

$$J^2 = L_1^2 + L_2^2 - 2 * L_1 * L_2 \cos(\beta) \quad \text{Ec.3.17.}$$

$$\text{sen}(\theta) = \frac{y}{J} \rightarrow y = J * \text{sen}(\theta) \quad \text{Ec.3.18.}$$

$$\text{cos}(\theta) = \frac{c}{J} \rightarrow C = J * \text{cos}(\theta) \quad \text{Ec.3.19.}$$

$$C = L_1 + L_2 - x \rightarrow x = L_1 + L_2 - C \quad \text{Ec.3.20.}$$

Con estas ecuaciones se consiguen obtener las trayectorias y ángulos de la pierna de pacientes con diferentes medidas, ya que, se observan los ángulos del paciente medido y con esto se halla la posición en X , y que debe alcanzar la pierna de diferentes pacientes.

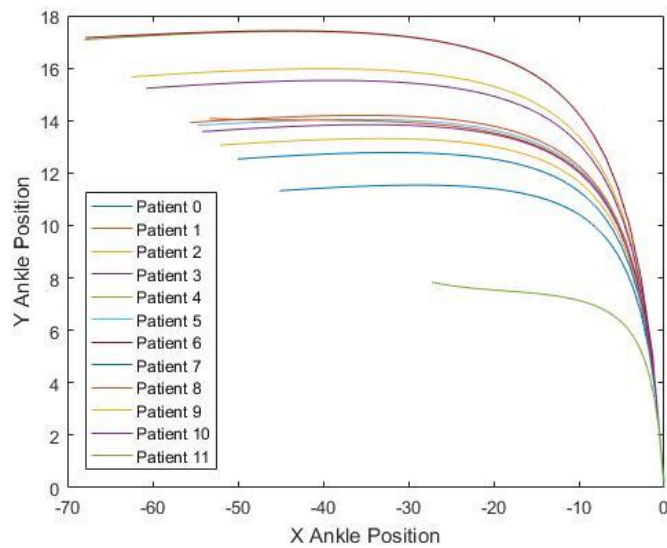
Estas trayectorias serán usadas para realizar la optimización del mecanismo, para este se usaron las medidas siguientes:

Tabla 1. Medidas antropométricas de pacientes

Paciente	Medida cadera – rodilla (cm)	Medida rodilla – tobillo (cm)
1	40	40
2	45	45
3	43,5	44
4	50	48
5	41	38
6	51	47
7	33	32
8	35	43
9	37	38
10	39	39
11	24	17

Las medidas anteriores fueron utilizadas después de hacer los ejercicios con el paciente control para calcular la trayectoria en cada paciente, partiendo de las medidas antropométricas de 11 sujetos, con lo cual se comprueba que el algoritmo puede calcular la trayectoria para cada paciente.

Ilustración 8. Trayectorias calculadas



3.1. Mecanismo que imita la trayectoria en los movimientos de flexo-extensión.

Una vez obtenidas las trayectoria de los pacientes que conformaron el grupo experimental, que en este caso fueron 12 incluyendo el paciente de referencia; se consolida el área de trabajo del mecanismo y de esta forma se procede a realizar la síntesis de mecanismo que se divide en tres partes: síntesis de número, síntesis de tipo y síntesis dimensional, en donde se analiza el mecanismo sin tener en cuenta la pierna y el mecanismo teniendo en cuenta la pierna.

El mecanismo que se va a utilizar para llevar a cabo la construcción es el mecanismo que se calcula teniendo en cuenta la pierna como parte de la cinemática del robot. Los cálculos hechos con el mecanismo sin tener en cuenta la pierna se realizan para generar un paralelo de los dos mecanismos y para hacer una explicación más detallada de cada uno de los métodos aplicados para el diseño del robot, de esta forma, se pretende que el mecanismo sin tener en cuenta la pierna pueda servir de ejemplo para aprender a usar las diferentes metodologías usadas para el diseño.

Síntesis del mecanismo:

1. Mecanismo sin tener en cuenta la pierna.

a. Síntesis de número

M = Variable que identifica la movilidad de los mecanismos, esto se trata de la trayectoria a seguir, si esta es una línea tiene un valor de 1, si es un área tiene un valor de 2 y si es en el espacio tiene un valor de 3. Cuando se trata de una curva su valor es 1, cuando se trata de un área de trabajo su valor es 2.

λ = Espacio de trabajo, cuando se trata de un plano su valor es 3, cuando se trata del espacio su valor es 6.

V = Número de circuitos del mecanismo, en este caso depende del diseñador colocar el número de circuitos que desea para el mecanismo, esta variable puede ir modificándose con el fin de encontrar el mejor resultado.

$$M = 2$$

$$\lambda = 3$$

$$V = 1$$

Se halla la movilidad con la ecuación 4.1, de esta ecuación se despeja j (número de juntas).

$$M = -\lambda * V + j \quad \text{Ec.4.1.}$$

Despejando j se encuentran el número de juntas en el mecanismo.

$$j = \text{Número de juntas}$$

$$j = M + \lambda V \quad \text{Ec.4.2.}$$

$$j = 2 + (3 * 1) \quad \text{E.c.4.3.}$$

$$j = 5$$

Se halla el número de circuitos con la ecuación 4.4, de esta se despeja n (número de eslabones del mecanismo).

$$-V = n - 1 - j \quad \text{Ec.4.4.}$$

Despejando n se encuentran el número de eslabones a tener en el mecanismo.

$$n = -V + 1 + j$$

$$n = -1 + 1 + 5 \quad \text{Ec.4.5.}$$

$$n = 5$$

Se halla el número de conexiones del mecanismo e .

$$e = j * 2 \quad \text{Ec.4.6.}$$

$$e = 5 * 2 = 10 \quad \text{Ec.4.7.}$$

b. Síntesis de tipo

En esta parte de la síntesis de mecanismos se hará el análisis de los valores obtenidos y se obtendrá el número de eslabones a usar y su tipología.

En la ilustración 9 se muestran los eslabones que debe tener el mecanismo según los cálculos hechos $n = 5$. Los eslabones son representados por cada una de las figuras de color verde, como se puede ver por el valor que se obtuvo, el mecanismo va a tener 5 eslabones.

Ilustración 9. Síntesis de tipo



En la ilustración 10 se muestran las conexiones con la tipología de los eslabones a usar en el mecanismo $e = 10$. La tipología de cada eslabón

Ilustración 10. Síntesis de tipo

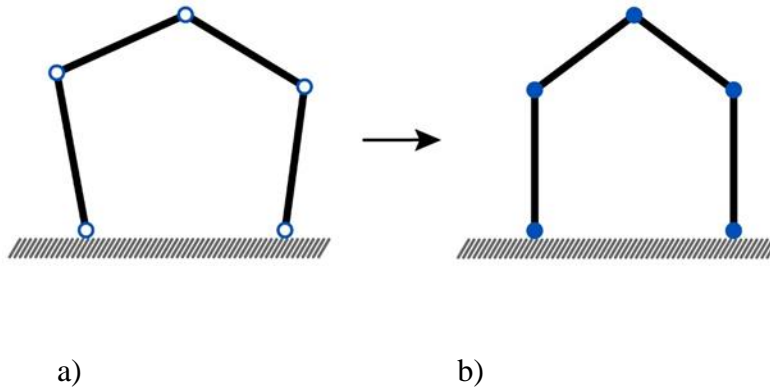


Debido a que no se pueden formar eslabones ternarios, cuaternarios u otros, debido a que el número de conexiones sólo permiten eslabones binarios y por lo tanto anclar cualquier eslabón a tierra daría como resultado el mismo mecanismo.

En la ilustración 11 se muestra el mecanismo observando que en la figura a) se observa el esquema encontrado al conectar los eslabones hallados con su topología y colocando cada junta como juntas de revolución, en la figura b) vemos el mecanismo ya implementado teniendo en cuenta que este sería un mecanismo de 5 barras. Sólo se encuentra esta configuración apta para el diseño de un mecanismo funcional.

En este caso los actuadores se encuentran en las dos juntas que van ancladas a la base.

Ilustración 11. Formación de eslabones



c. Síntesis dimensional

La síntesis dimensional puede ser clasificada de tres tipos:

- Generación de función
- Generación de camino
- Generación de movimiento

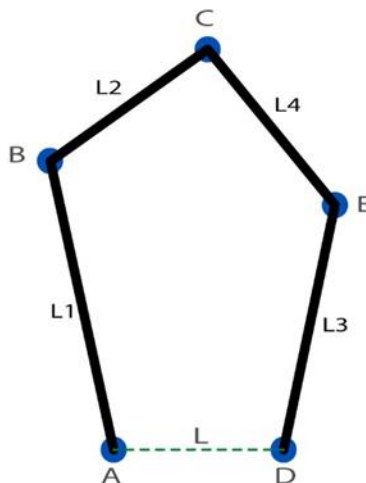
En este caso se determina usar generación de camino debido a que se pretende que el mecanismo pueda seguir una serie de trayectorias que se hallaron al realizar las prácticas de rehabilitación por parte del fisioterapeuta a cargo.

Para esto se hace necesario tener la cinemática inversa y directa del mecanismo a usar.

Cinemática inversa

La cinemática inversa se halla con el fin de conocer el ángulo de los actuadores del mecanismo de 5 barras requerido para que el mecanismo se ubique en la posición conocida.

Ilustración 12. Cinemática inversa



Teniendo en cuenta la ilustración 12 se hallan las normas de los vectores correspondientes a los puntos \overrightarrow{AC} y \overrightarrow{DC} . Teniendo en cuenta que los actuadores se situarán en los puntos A y D.

$$a = \text{norm}(C - A)$$

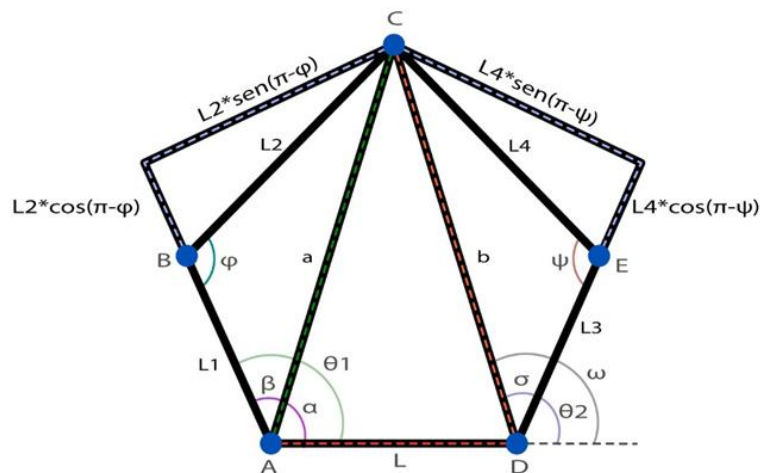
$$b = \text{norm}(C - D)$$

Se da va orientación de los eslabones L_1 y L_2 con la variable C_{d1} y C_{d2} respectivamente

$$C_{d1} = 1$$

$$C_{d2} = 1$$

Ilustración 13. Análisis geométrico mecanismo de 5 barras sin punto de interés



Se hallan los ángulos correspondientes a la orientación del punto C con respecto a la horizontal en los puntos A y D.

$$\alpha = \text{atan2}(C_y, C_x) \quad \text{Ec.4.8.}$$

$$w = \text{atan2}(C_y, C_x - L) \quad \text{Ec.4.9.}$$

Utilizando la ley de los cosenos se halla el ángulo φ .

$$\cos(\varphi) = (L_1^2 + L_2^2 + a^2) / (2 * L_1 * L_2) \quad \text{Ec.4.10.}$$

$$\text{sen}(\varphi) = C_d * \sqrt{1 - \cos(\varphi)^2} \quad \text{Ec.4.11.}$$

$$\varphi = \text{atan2}(\text{sen}(\varphi), \cos(\varphi)) \quad \text{Ec.4.12}$$

Utilizando la ley de los cosenos se halla el ángulo Ψ .

$$\cos(\psi) = ((L_3^2 + L_4^2 + b^2) / (2 * L_3 * L_4)) \quad \text{Ec.4.13.}$$

$$\text{sen}(\psi) = C_d * \sqrt{1 - \cos(\psi)^2} \quad \text{Ec.4.14.}$$

$$\Psi = \text{atan2}(\text{sen}(\psi), \cos(\psi)) \quad \text{Ec.4.15.}$$

Se hallan los ángulos que corresponden a β y σ .

$$\beta = \text{atan2}(L_4 * \text{sen}(\pi - \varphi), L_1 + L_2 * \cos(\pi - \varphi)) \quad \text{Ec.4.16.}$$

$$\sigma = \text{atan2}(L_4 * \text{sen}(\pi - \psi), L_3 + L_4 * \cos(\pi - \psi)) \quad \text{Ec.4.17.}$$

Teniendo todos los ángulos se hallan θ_1 y θ_2 .

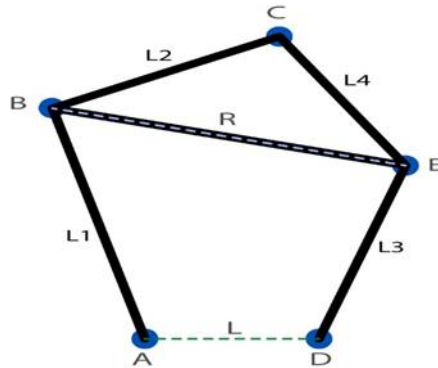
$$\theta_1 = \alpha + \beta \quad \text{Ec.4.18.}$$

$$\theta_2 = w - \sigma \quad \text{Ec.4.19.}$$

Cinemática directa

La cinemática directa se halla con el fin de conocer la posición del efector final conociendo los ángulos de los actuadores del mecanismo.

Ilustración 14. Cinemateca directa



Se hallan los valores de cada uno de los puntos A, B, C, D, E en sus componentes del mecanismo.

$$A_x = 0$$

$$A_y = 0$$

$$B_x = L_1 * \cos(\theta_1) + A_x \quad \text{Ec.4.20.}$$

$$B_y = L_1 * \text{sen}(\theta_1) + A_y \quad \text{Ec.4.21.}$$

$$D_x = L + A_x \quad \text{Ec.4.22.}$$

$$D_y = A_y$$

$$E_x = L_3 * \cos(\theta_2) + L + A_x \quad \text{Ec.4.23.}$$

$$E_y = L_3 * \text{sen}(\theta_2) + A_y \quad \text{Ec.4.24.}$$

Se halla el valor de R y θR .

$$R = \sqrt{(L - L_1 * \cos(\theta_1) + L_3 * \cos(\theta_2))^2 + (L_1 * \text{sen}(\theta_1) - (L_3 * \text{sen}(\theta_2)))^2} \quad \text{Ec.4.25.}$$

$$\theta R = \text{atan2}((L_3 * \text{sen}(\theta_2) - L_1 * \text{sen}(\theta_1)), L - L_1 * \cos(\theta_1) + L_3 * \cos(\theta_2)) \quad \text{Ec.4.26.}$$

$$\theta_{12} = \theta R + \text{acos}((R^2 + L_2^2 - L_4^2) / (2 * R * L_2)) \quad \text{Ec.4.27.}$$

Se hallan las componentes en x y y del efector final C

$$C_x = B_x + L_2 * \cos(\theta_{12}) \quad \text{Ec.4.28.}$$

$$C_y = B_y + L_2 * \text{sen}(\theta_{12}) \quad \text{Ec.4.29.}$$

2. Mecanismo teniendo en cuenta la pierna.

A diferencia del análisis hecho en el mecanismo anterior, en este análisis se tiene en cuenta la pierna del paciente como una extensión del mecanismo. Esto debido a que la pierna va a ser sujeta el efector final del mecanismo cierra un circuito y genera una cadena cinemática cerrada con dos o más circuitos dependiendo del mecanismo que se dé como resultado.

a. Síntesis de número

$$M = 2$$

$$\lambda = 3$$

$$V = 2$$

Se halla la movilidad con la ecuación 4.30, de esta ecuación se despeja j (número de juntas)

$$M = -\lambda V + j \quad \text{Ec.4.30.}$$

Despejando j se encuentran el número de juntas en el mecanismo.

$$j = M + \lambda V \quad \text{Ec.4.31.}$$

$$j = 2 + (3 * 2) \quad \text{Ec.4.32.}$$

$$j = 8$$

Se halla el número de circuitos con la ecuación 4.33, de esta se despeja n (número de eslabones del mecanismo).

$$-V = n - 1 - j \quad \text{Ec.4.33.}$$

$$n = -V + 1 + j$$

$$n = -2 + 1 + 8 \quad \text{Ec.4.34.}$$

$$n = 7$$

Se halla el número de conexiones del mecanismo e .

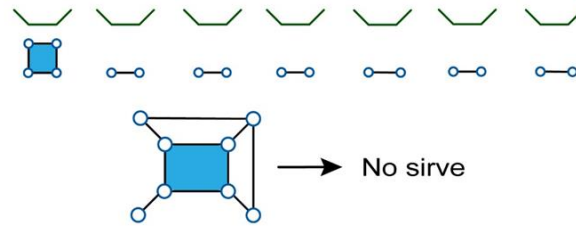
$$e = j * 2 \quad \text{Ec.4.35.}$$

$$e = 8 * 2 = 16 \quad \text{Ec.4.36.}$$

b. Síntesis de tipo

$$n = 7 \quad e = 16$$

Ilustración 15. Síntesis de tipo (error)

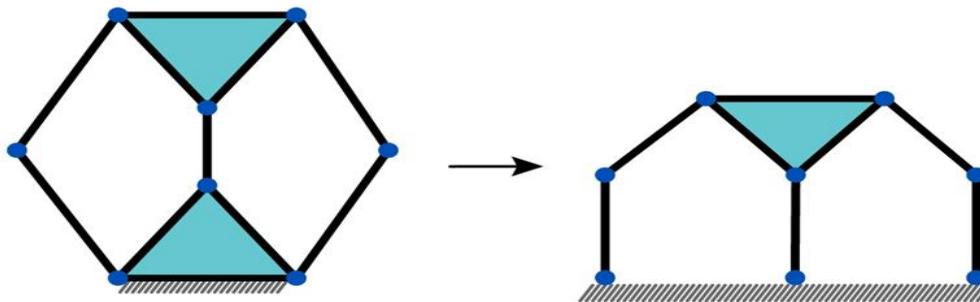


Ninguna junta puede quedar sin conectarse a un eslabón por lo tanto esta liquidación no es realizable.

Ilustración 16. Síntesis de Tipo (correcto)



Ilustración 17. Mecanismo con dos circuitos



En este caso podemos observar un mecanismo con dos circuitos el cual contiene en su parte derecha el mecanismo que se calculó para el área de trabajo y en su parte izquierda lo que sería la pierna del paciente a rehabilitar.

c. Síntesis dimensional

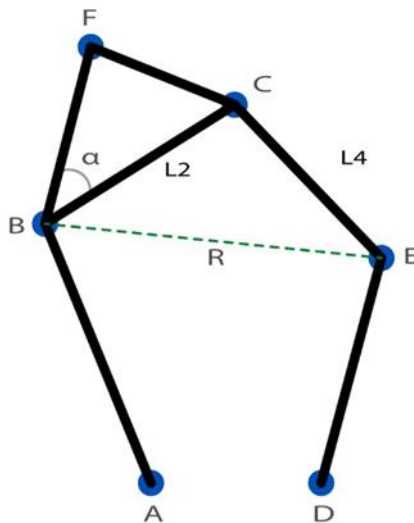
En este caso se determina usar generación de camino debido a que se pretende que el mecanismo pueda seguir una serie de trayectorias que se hallaron al realizar las prácticas de rehabilitación por parte del fisioterapeuta a cargo.

Para esto se hace necesario tener la cinemática inversa y directa del mecanismo a usar.

Cinemática inversa

La cinemática inversa se halla con el fin de conocer el ángulo de los actuadores del mecanismo requerido para que el mecanismo se ubique en la posición conocida.

Ilustración 18. Cinemática inversa



Se halla la variable C_d para definir la orientación del mecanismo.

$$C_d = 1$$

Se halla la norma del vector \overrightarrow{AF} .

$$\alpha = \text{norm}(F - A) \quad \text{Ec.4.37.}$$

Se halla α .

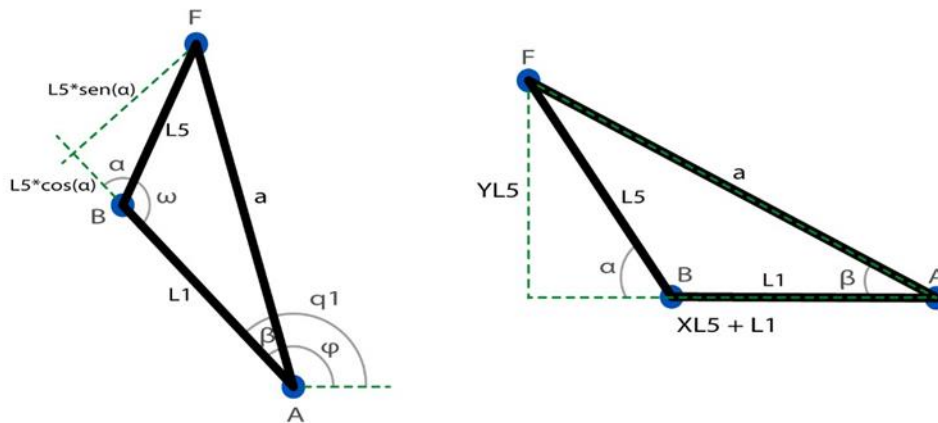
$$\cos(\alpha) = \frac{a^2 - L_1^2 - L_5^2}{2 * L_1 * L_5}; \quad \cos(180 - w) \quad \alpha = 180 - w \quad \text{Ec.4.38}$$

$$\text{sen}(\alpha) = \text{atan2}(\text{sen}(\alpha), \cos(\alpha))$$

$$X L_5 = L_5 * \cos(\alpha) \quad \text{Ec.4.39.}$$

$$Y L_5 = L_5 * \text{sen}(\alpha) \quad \text{Ec.4.40}$$

Ilustración 19. Análisis geométrico del mecanismo de 5 barras con punto de interés 1



Se halla el ángulo β con los lados de los triángulos auxiliares.

$$\beta = \text{atan2}(Y L_5, X L_5 + L_1) \quad \text{Ec.4.41.}$$

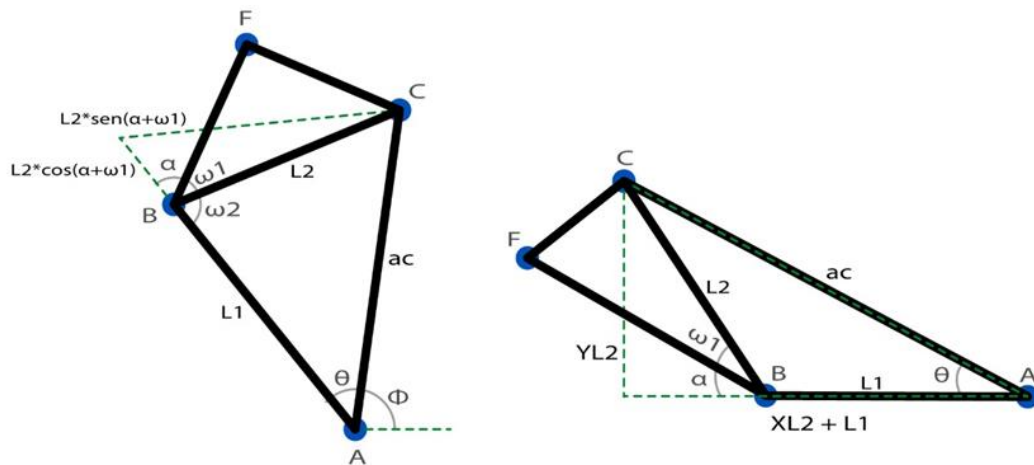
$$q_1 = \beta + (\text{atan2}(F_y, F_x)) \quad \text{Ec.4.42.}$$

$$Y L_2 = L_2 * \text{sen}(\alpha + \omega_1) \quad \text{Ec.4.43.}$$

$$XL_2 = L_2 * \cos(\alpha + \omega_1)$$

Ec.4.44.

Ilustración 20. Análisis geométrico del mecanismo de 5 barras con punto de interés 2



Se halla la magnitud de \overline{AC}

$$ac = \frac{\sqrt{(L_2 * \cos(\alpha + \omega_1) + L_1)^2 + (L_2 * \text{sen}(\alpha + \omega_1))^2}}{(L_2 * \text{sen}(\alpha + \omega_1))^2}$$

Ec. 4.45.

Se halla el ángulo θ .

$$\theta = \text{atan2}(L_2 * \text{sen}(\alpha + \omega_1), L_2 * \cos(\alpha + \omega_1) + L_1)$$

Ec.4.46.

Se halla el ángulo ϕ .

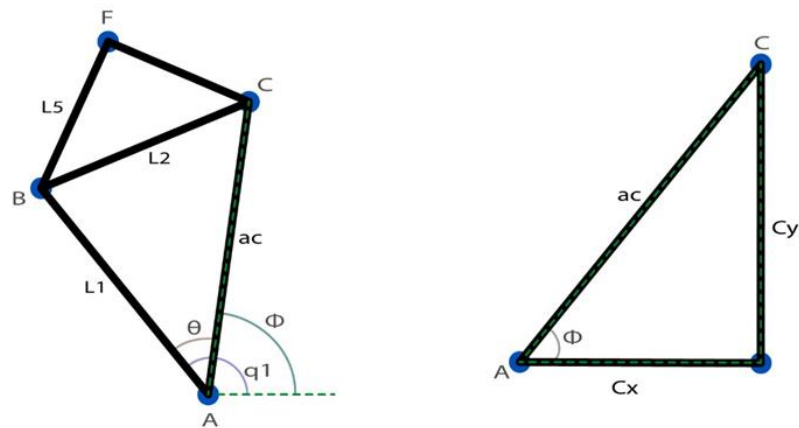
$$\phi = q_1 - \theta$$

Ec.4.47.

Con lo anterior se puede hallar la posición en x y y del punto C .

$$C_x = ac * \cos(\phi) \quad C_y = ac * \text{sen}(\phi) \quad \text{Ec.4.48.}$$

Ilustración 21. Análisis geométrico del mecanismo de 5 barras con punto de interés 3



Teniendo la posición del punto C se coloca la orientación del mecanismo en este punto mediante la variable C_d2 .

$$C = [C_x, C_y] \quad C_d2 = 1$$

Se halla la norma del vector \overrightarrow{DC} .

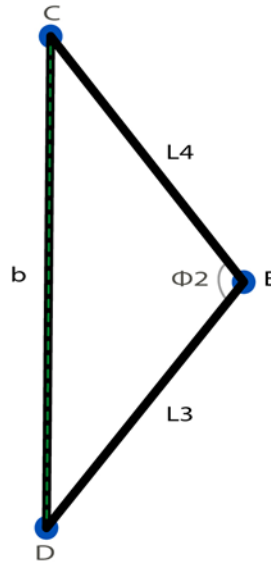
$$B = \text{norm}(C - D) \quad \text{Ec.4.49.}$$

Se halla el ángulo $\phi2$.

$$\cos(\phi_2) = \frac{L_3^2 + L_4^2 - b^2}{2 * L_3 * L_4} \quad \text{Ec.4.50.}$$

$$\text{sen}(\phi_2) = C_d 2 * \sqrt{1 - \cos(\phi_2)^2} \quad \text{Ec.4.51.}$$

Ilustración 22. Análisis geométrico del mecanismo de 5 barras con punto de interés 4



$$\phi_2 = \text{atan2}(\text{sen}(\phi_2), \cos(\phi_2)) \quad \text{Ec.4.52.}$$

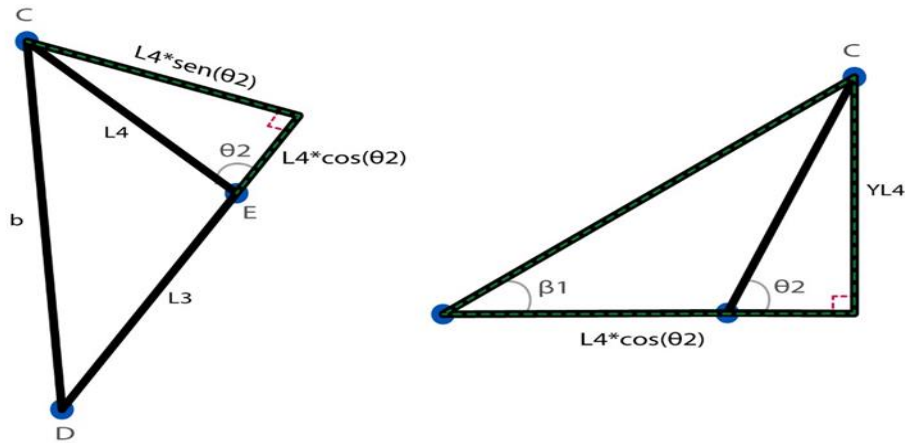
Se halla el ángulo θ_2 .

$$\cos(\text{tetta2}) = \frac{b^2 - L_3^2 - L_4^2}{2 * L_3 * L_4} \quad \theta_2 = 180 - \phi_2 \quad \text{Ec.4.53.}$$

$$\text{sen}(\text{tetta2}) = C_d 2 * \sqrt{1 - \cos(\text{tetta2})^2} \quad \text{Ec.4.54.}$$

$$\theta_2 = \text{atan2}(\text{sen}(\text{tetta2}), \cos(\text{tetta2})) \quad \text{Ec.4.55.}$$

Ilustración 23. Análisis geométrico del mecanismo de 5 barras con punto de interés 5



Usando los triángulos auxiliares se halla q_2

$$YL_4 = L_4 \text{ sen } (\phi_2) \quad XL_4 = L_4 * \cos(\phi_2) \quad \text{Ec.4.56.}$$

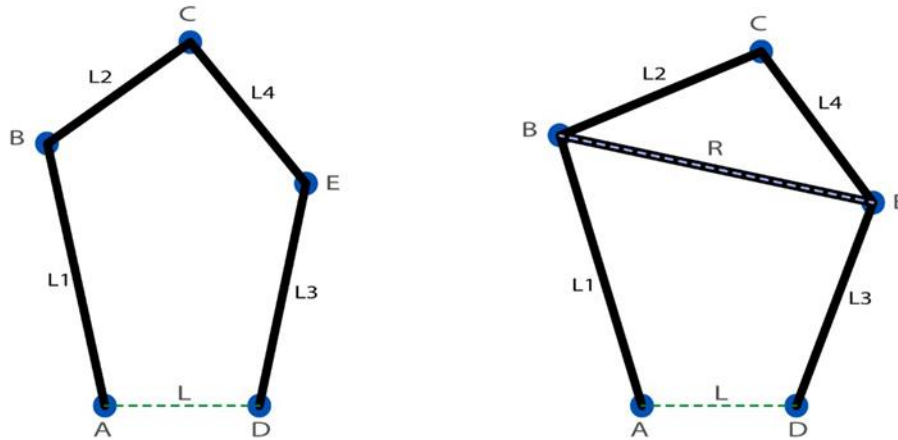
$$\beta_1 = \text{atan2} (L_4 * \text{sen } (\phi_2), (L_4 * \cos(\phi_2) + L_3) \quad \text{Ec.4.57.}$$

$$q_2 = \text{atan2} ((C_y - A_y), ((x - A_x)) - \beta_1 \quad \text{Ec.4.58.}$$

Cinemática Directa

La cinemática directa se halla con el fin de conocer la posición del efector final conociendo los ángulos de los actuadores del mecanismo.

Ilustración 24. Cinemática directa



Se hallan las posiciones de los puntos del mecanismo.

$$A_x = 0 \quad A_y = 0$$

$$B_x = L_1 * \cos(Q_1) + A_x \tag{Ec.4.59.}$$

$$B_y = L_1 * \text{sen}(Q_1) + A_y \tag{Ec.4.60.}$$

$$D_x = L + A_x \tag{Ec.4.61.}$$

$$E_x = L_2 * \cos(Q_2) + L + A_x \tag{Ec.4.62.}$$

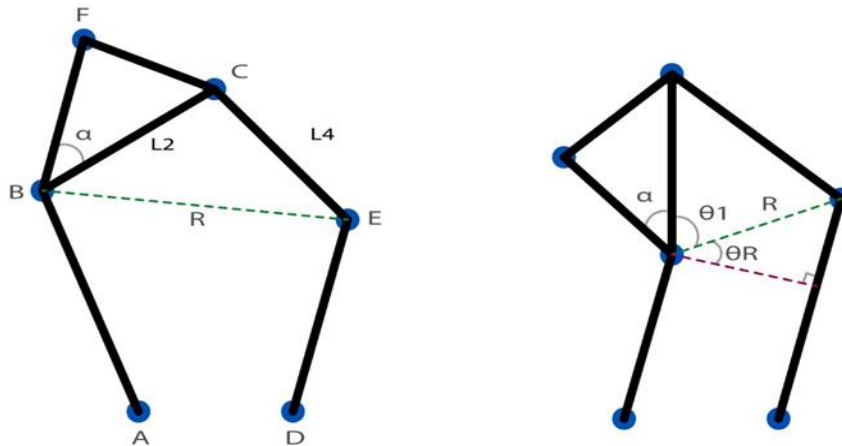
$$E_y = L_3 * \text{sen}(Q_2) + A_y \tag{Ec.4.63.}$$

Se halla R y θ_R .

$$R = \sqrt{(L - L_1 * \cos(q_1) + L_3 * \cos(q_2))^2 + (-L_1 * \text{sen}(q_1) + L_3 * \text{sen}(q_2))^2} \tag{Ec.4.64.}$$

$$\theta_R = \text{atan2}(L_3 * \text{sen}(q_2) - L_1 * \text{sen}(q_1), L - L_1 * \cos(q_1) + L_3 * \cos(q_2)) \tag{Ec.4.65.}$$

Ilustración 25. Análisis geométrico del mecanismo de 5 barras con punto de interés 6



Se halla θ_1 y θ_{12} .

$$\theta_1 = \text{acos}((R^2 + L_2^2 - L_4^2) / (2 * R * L_2)) \quad \text{Ec.4.66.}$$

$$\theta_{12} = \text{acos}((R^2 + L_2^2 - L_4^2) / (2 * R * L_2)) \quad \text{Ec.4.67.}$$

$$\theta_{12} = \theta_R + \theta_1 \quad \text{Ec.4.68.}$$

Teniendo en cuenta los ángulos anteriores se hallan las posiciones de los puntos C y F .

$$Cx = Bx + L_2 * \cos(\theta_{12}) \quad \text{Ec.4.69.}$$

$$Cy = By + L_2 * \text{sen}(\theta_{12}) \quad \text{Ec.4.70.}$$

$$Fx = Bx + L_5 * \cos(\theta_{12} + \alpha) \quad \text{Ec.4.71.}$$

$$Fy = By + L_5 * \text{sen}(\theta_{12} + \alpha) \quad \text{Ec.4.72.}$$

Optimización del mecanismo

Para la optimización del mecanismo se consideraron diferentes técnicas de inteligencia artificial, las cuales fueron algoritmos genéticos y enjambre de partículas, este último fue el que se trabajó para la optimización del mecanismo de 5 barras con punto de interés debido a que este mostró ser más eficaz a la hora de dar solución a este problema. A continuación se detalla el funcionamiento del algoritmo de enjambre de partículas (PSO).

$$w = \left[\frac{w_{min} - w_{max}}{(iter_{max} - 1)} (iter - 1) \right] + w_{max} \quad \text{Ec.4.74.}$$

C1, C2 constantes de aceleración.

C1 > 0 y C2 = 0 partículas independientes (importa más el comportamiento individual).

C1 = 0 y C2 > 0 partículas colectivas (importa más el comportamiento grupal)

C1 = C2 > 0 Partículas atraídas por valor promedio.

C1 > C2 experiencia propia mejor que la de grupo.

C1 < C2 experiencia de grupo mejor que la propia.

C1 y C2 bajas trayectorias de desplazamiento suave.

C1 y C2 altas movimientos abruptos.

$$C1 = \frac{(C1_{min}-C1_{max})}{iter_{max}} iter + C1_{max} \quad \text{Ec.4.75.}$$

Pasa de un valor grande a uno pequeño con el tiempo

$$C2 = \frac{(C2_{max}-C2_{min})}{iter_{max}} iter + C2_{min} \quad \text{Ec.4.76.}$$

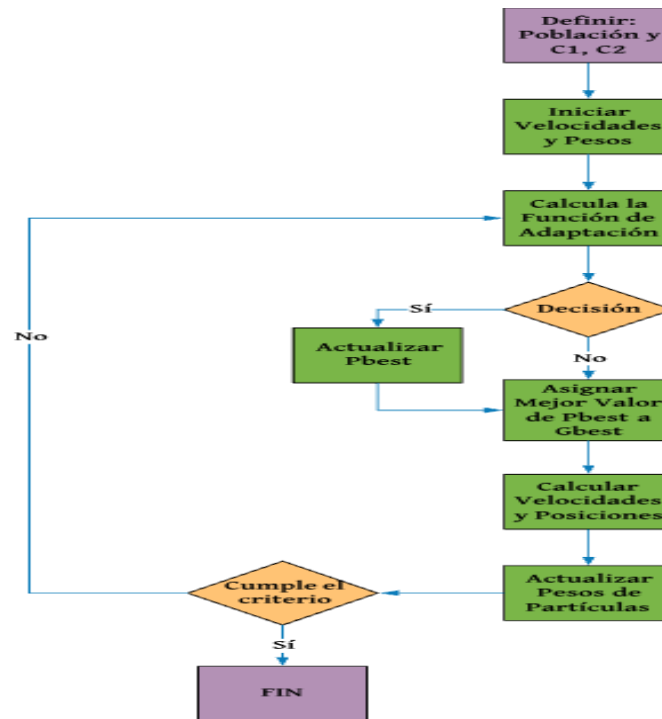
Pasa de un valor pequeño a uno grande con el tiempo.

Se actualiza la posición cambiando los pesos de las partículas decidiendo de esta forma cuál es la mejor para seguir, haciendo que la optimización se vaya actualizando y hallando las mejores condiciones para llegar a la respuesta deseada [19].

$$X_i(t + 1) = X_i(t) + V_i(t + 1) \quad \text{Ec.4.77.}$$

En la ilustración 26 se aprecia un diagrama de flujo del proceso de optimización. En este se explica gráficamente el paso a paso de cómo funciona el PSO, empezando por definir la población, después se dan indicaciones de los pesos, se calcula la función que determina si el error se va disminuyendo indicando que se está encontrando una solución, se encuentran cambian los valores de los pesos de cada partícula indicando cuál de estas se acerca a la repuesta, se actualizan los pesos de las partículas y se reinicia el procesos hasta tanto se encuentre la mejor solución. Este proceso se aplica para cada una de la funciones objetivo que se presentan en la siguiente sección.

Ilustración 26. Optimización por enjambre de partículas



Para realizar la optimización con Matlab, primero se tuvo que declarar los límites de las variables a optimizar en este caso son:

L = Distancia entre los ejes de los motores.

L_1 y L_3 = El largo de los eslabones que van conectados a los motores, que para el caso se deja una sola variable ya que deben ser iguales.

L_2 y L_4 = El largo de los eslabones que se unen en la punta del mecanismo de 5 barras, para el caso se deja una sola variable ya que deben ser iguales.

D_x = Distancia en x de la base de referencia del motor con respecto a la base de referencia de la trayectoria.

En este caso se optimizó el mecanismo de 5 barras sin el punto de interés debido al número de variables y la complejidad que esta representa.

En esta optimización se debe tener una función a la cual se le envía las 5 variables antes descritas entre los rangos dados, esto por medio de un vector, estos valores son los que se irán modificando hasta lograr el funcionamiento indicado.

La función que se denomina *fitnes function* toma los valores enviados por el algoritmo y este contiene la cinemática inversa y directa, así como las trayectorias a realizar. De esta manera la función comprueba matemáticamente si el mecanismo puede realizar o no todas las trayectorias que se establecieron para conformar un área de trabajo.

En el caso de no poder realizar alguna de las trayectorias completamente este enviará al algoritmo de optimización un valor en este caso de $10 * 10^{10}$ esto, ya que, se le quiere dar a entender al algoritmo que los valores asignados son erróneos. En caso de que las variables pueden desarrollar todas las trayectorias se enviará un valor equivalente a:

Valor devuelto al algoritmo: $L + 2 * L_1, L_3 + 2 * L_2, L_4$ haciendo así entender al algoritmo que los valores dados satisfacen lo pedido y que debe ir en esa dirección según su funcionamiento como se describió anteriormente, de esta forma, se hallan las dimensiones más pequeñas posibles del mecanismo.

Después de hacer esta optimización se realiza la siguiente la cual se encargará de hallar el punto de interés del mecanismo, en este caso se tomarán en cuenta los valores obtenidos de L, L_1, L_3, L_2, L_4 y se hallarán los siguientes:

L_5 = Distancia del eslabón L_1 hasta el punto de interés.

D_x = Distancia en x con punto de interés.

D_y = Distancia en y con punto de interés.

Y de esta forma se completa la optimización del mecanismo. En estas optimizaciones se tuvieron en cuenta las variables dimensionales, haciendo del mecanismo lo más pequeño posible.

Los valores de las dimensiones del mecanismo obtenidos en la optimización podrán verse en la sección de resultados.

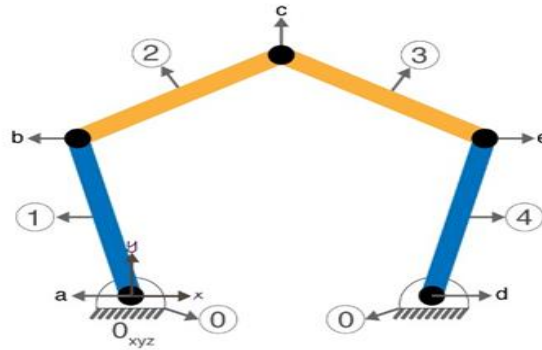
Aplicación del método de Davies

a. Mecanismo sin pierna (Mecanismo de 5 barras)

El mecanismo está compuesto por $n = 5$ cuerpos y por $e = 5$ acoples, se enumeran las barras colocando la barra fija como (0) y se siguen numerando todos los cuerpos, se elige el valor de $\lambda = 3$ esto debido a que se trabaja en el plano.

Etapa 1a. En la etapa inicial se ilustra la representación esquemática del mecanismo. Es una versión simplificada del modelo físico conteniendo las informaciones de topología y geometría basado del sistema inercial de coordenadas O_{XYZ} .

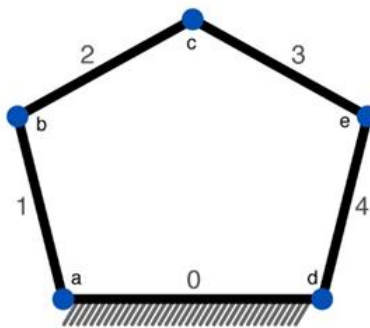
Ilustración 27. Representación esquemática del mecanismo de 5 barras y sistema inercial O_{XYZ}



La localización del origen O_{XYZ} puede ser arbitraria pero debe ser coherente minimizando el problema, en este caso O_Z es el eje paralelo de los pares rotativos.

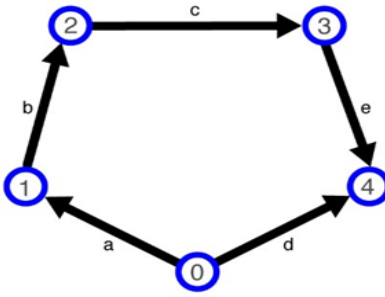
Etapas 1b. se forma la red de acoples del mecanismo. Básicamente es una representación de la topología del mecanismo donde los cuerpos son polígonos cuyo número de vértices es el grado de conexión (número de acoples) en cada cuerpo.

Ilustración 28. Red de acoples del mecanismo



Etapas 1c. Esta etapa trata de la construcción del grafo de acoples G_C de la red, donde cada cuerpo es representado por una arista, como lo muestra la siguiente figura.

Ilustración 29. Grafo de acoples G_C del mecanismo.



En el grafo de acoples G_C sus aristas son orientadas en el sentido de los vértices menores para los mayores.

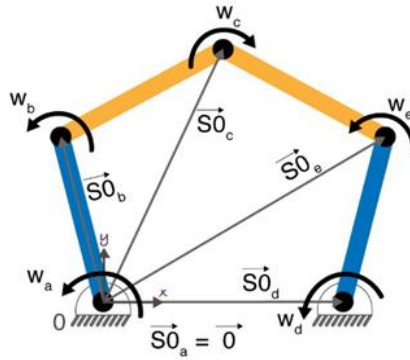
La etapa 1 es igual tanto para la cinemática como para la estática, por esta razón a partir de la Etapa 2 se hacen los análisis por separado.

Cinemática del mecanismo sin pierna (Mecanismo de 5 barras)

Etapa 2a. En esta etapa se reúnen las características de los acoples necesarios para la formación de los heligros y la construcción del grado de movimientos G_M , las principales informaciones son: el vector de dirección del eje \vec{S}^M el vector posición \vec{S}_0 en relación al origen de O_{XYZ} y el paso h^M . La información relevante al grafo son los movimientos unitarios fi de cada acople. Determinar el tipo de par cinemático de los acoples que son fundamentales para conocer estas propiedades.

Todos los acoples (a, b, c, d, e) poseen el eje de sus heligros paralelos al eje O_z , los vectores de posición \vec{S}_0 de los heligros y los movimientos en los acoples son presentados en la siguiente figura.

Ilustración 30. Vectores de posición \vec{S}_0 y los respectivos movimientos en el mecanismo.



Los acoples de tipo par rotativo poseen como característica que $f = 1$ en el espacio, permitiendo apenas una rotación (w). Por lo tanto, para todos los heligros de cada acople, el vector de dirección unitario es paralelo a O_z :

$$\vec{S}_Z^M = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{Ec.4.78.}$$

Donde z describe la dirección del movimiento unitario.

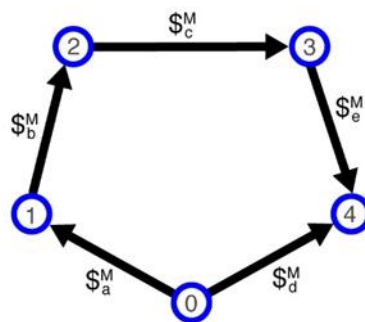
Los vectores posición \vec{S}_0 de los heligros de los acoples (a, b, c, d, e) son descritos como

$$\vec{S}_{0a} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \vec{S}_{0b} = \begin{pmatrix} b_x \\ b_y \\ 0 \end{pmatrix} \quad \vec{S}_{0c} = \begin{pmatrix} c_x \\ c_y \\ 0 \end{pmatrix} \quad \vec{S}_{0d} = \begin{pmatrix} d_x \\ 0 \\ 0 \end{pmatrix} \quad \vec{S}_{0e} = \begin{pmatrix} e_x \\ e_y \\ 0 \end{pmatrix} \quad \text{Ec.4.79.}$$

Los pasos de los heligeros son nulos ($h^M = 0$) puesto que hay rotación pura.

Etapas 3a. Una vez que se conoce la geometría y la topología del mecanismo se debe generar un grafo de movimiento G_M . Esto se realiza sustituyendo las aristas de cada acople en G_C por $f = 1$ aristas de serie en G_M , manteniendo el sentido preexistente. Así es posible destacar los movimientos unitarios de cada acople, permitiendo hallar el grado de libertad total (F) del mecanismo.

Ilustración 31. Grafo de movimientos G_M del mecanismo.



En el caso de este mecanismo los grafos G_C y G_M son idénticos pues todos los $e = 5$ acoples poseen un único movimiento unitario ($f = 1$) consecuente con el grado de libertad total (F) es

$$F = \sum_{i=1}^e f_i = 1 + 1 + 1 + 1 + 1 = 5 \quad \text{Ec.4.80.}$$

Etapas 3b. Esta etapa se considera como una de las más importantes del método, ya que, la conversión de las informaciones topológicas contenidas en el grafo G_M para la forma matricial,

resultando en la matriz de los f circuitos $[B] \times F$. Para esto es necesario identificar los f circuitos.

Primero se determina el número de f circuitos (1)

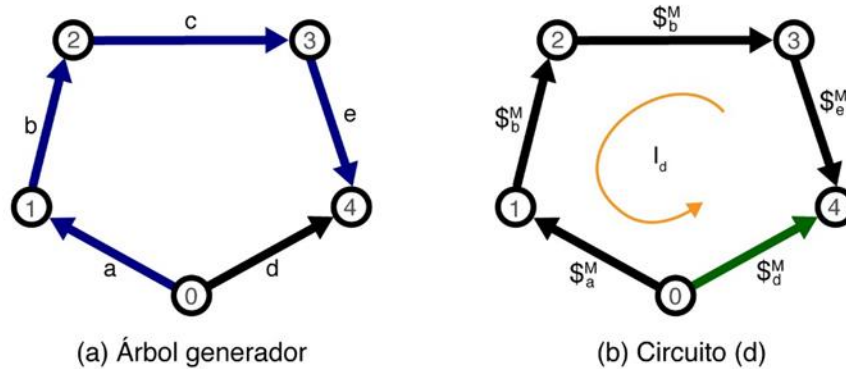
$$l = e - n + 1 = 5 - 5 + 1 = 1$$

Claramente solo existe un circuito en el grafo G_M . Para otros mecanismos topológicamente más complejos es importante realizar este cálculo.

Conociendo el número de f circuitos, se debe determinar a través de la selección de un árbol generador.

Para este mecanismo existen 5 posibilidades de árbol generador y se obtiene al seleccionar la arista del acople (d) como cuerda. De esta forma el árbol generador resultante contempla las aristas (a , b , c y e). El f circuito identificado va en sentido de la cuerda (d).

Ilustración 32. Determinación del f circuito del mecanismo.



La matriz de f circuitos del mecanismo se toma:

a b c e d

$$[B_M] \text{ } 1 \times 5 = [B_M] \text{ } 1 \times 5 = [-1 \ -1 \ -1 \ -1 \ 1] \text{ } 1 \times 5 \quad \text{Ec. 4.81.}$$

La fila de la matriz representa el f circuito (d), en este caso sólo hay un circuito con 5 aristas por lo que es una matriz 1×5 , las columnas de la matriz representa las aristas (a, b, c, d, e). Solo el elemento de la columna (d) tiene signo positivo, ya que, es la única que va en el mismo sentido del circuito (d).

Etapas 4a. Dadas las características geométricas de los acoples se deben construir los heligiros. Con lo visto en las etapas anteriores todos los acoples de este mecanismo son el tipo par rotativo, indicando que los pasos de los heligiros son nulos ($h^M = 0$).

Los heligiros se construyen de la siguiente forma donde w hace referencia a las velocidades angulares en cada una de las aristas:

$$S_a^M = \begin{bmatrix} \dots & \vec{S}_a^M & \dots \\ \vec{S}_{oa} & x & \vec{S}_a^M \end{bmatrix} \quad w_a = \begin{matrix} 0 \\ 0 \\ w_a \\ \dots \\ 0 \\ 0 \\ 0 \end{matrix} \quad \text{Ec.4.82.}$$

$$\begin{aligned} \vec{\phi}_b^M &= \begin{bmatrix} \dots & \vec{S}_b^M & \dots \\ \vec{S}_{ob} & x & \vec{S}_b^M \end{bmatrix} & w_b &= \begin{bmatrix} 0 \\ 0 \\ w_b \\ \dots \\ b_y w_b \\ -b_x w_b \\ 0 \end{bmatrix} \end{aligned} \quad \text{Ec.4.83.}$$

$$\begin{aligned} \vec{\phi}_c^M &= \begin{bmatrix} \dots & \vec{S}_c^M & \dots \\ \vec{S}_{oc} & x & \vec{S}_c^M \end{bmatrix} & w_c &= \begin{bmatrix} 0 \\ 0 \\ w_c \\ \dots \\ c_y w_c \\ -c_x w_c \\ 0 \end{bmatrix} \end{aligned} \quad \text{Ec.4.84.}$$

$$\begin{aligned} \vec{\phi}_d^M &= \begin{bmatrix} \dots & \vec{S}_c^M & \dots \\ \vec{S}_{od} & x & \vec{S}_c^M \end{bmatrix} & w_d &= \begin{bmatrix} 0 \\ 0 \\ w_d \\ \dots \\ 0 \\ -d_x w_d \\ 0 \end{bmatrix} \end{aligned} \quad \text{Ec.4.85.}$$

$$\begin{aligned} \vec{\phi}_e^M &= \begin{bmatrix} \dots & \vec{S}_c^M & \dots \\ \vec{S}_{oe} & x & \vec{S}_c^M \end{bmatrix} & w_e &= \begin{bmatrix} 0 \\ 0 \\ w_e \\ \dots \\ e_y w_e \\ -e_x w_e \\ 0 \end{bmatrix} \end{aligned} \quad \text{Ec.4.86.}$$

Observando los heligros se percibe que las coordenadas L, M, R son nulas, evidenciando que los movimientos permitidos en el mecanismo están restringidos a las coordenadas N, P, Q del espacio plano ($\lambda = 3$). Por lo tanto es posible reescribir los heligros como:

$$\begin{matrix} \$a^M = \begin{Bmatrix} Wa \\ \dots \\ 0 \\ 0 \end{Bmatrix} & \$b^M = \begin{Bmatrix} Wb \\ \dots \\ b_y W_b \\ -b_x W_b \end{Bmatrix} & \$c^M = \begin{Bmatrix} Wc \\ \dots \\ c_y W_c \\ -c_x W_c \end{Bmatrix} & \$d^M = \begin{Bmatrix} Wd \\ \dots \\ 0 \\ -d_x W_d \end{Bmatrix} & \$e^M = \begin{Bmatrix} We \\ \dots \\ e_y W_e \\ -e_x W_e \end{Bmatrix} \end{matrix} \text{ Ec.4.87.}$$

La fila encima de la línea punteada identifica a la rotación y la parte inferior que corresponde a la translación.

Etapla 4b. La matriz de los movimientos.

$[M_D]_{3 \times 5}$ Reúne todos los heligros del mecanismo.

$$[M_D]_{3 \times 5} = \begin{matrix} & W_a & W_b & W_c & W_e & W_d \\ \begin{matrix} 0 \\ 0 \end{matrix} & \begin{matrix} b_y w_b \\ -b_x w_b \end{matrix} & \begin{matrix} c_y w_c \\ -c_x w_c \end{matrix} & \begin{matrix} e_y w_e \\ -e_x w_e \end{matrix} & \begin{matrix} 0 \\ -d_x w_d \end{matrix} & \end{matrix} \quad 3 \times 5 \quad \text{Ec.4.88.}$$

Donde las columnas corresponden a los heligros $\$a^M, \$b^M, \$c^M, \$d^M, \$e^M$ respectivamente.

Al separar las magnitudes de los heligros es posible obtener la matriz de los movimientos unitarios $[\widehat{M}_D]_{3 \times 5}$

$$[\widehat{M}_D]_{3 \times 5} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & b_y & c_y & e_y & 0 \\ 0 & -b_x & -c_x & -e_x & -d_x \end{bmatrix} \quad 3 \times 5 \quad \text{Ec.4.89.}$$

El vector de las magnitudes de los movimientos $\{\vec{\phi}\}_{F \times 1}$ es formado por

$$\{\vec{\phi}\}_{5 \times 1} = \begin{Bmatrix} W_a \\ W_b \\ W_c \\ W_e \\ W_d \end{Bmatrix} \quad 5 \times 1 \quad \text{Ec.4.90.}$$

Conteniendo las variables cinemáticas del problema.

Etapa 5a. En esta etapa, las informaciones geométricas y topológicas son reagrupadas en un único elemento matemático, la matriz de movimientos unitarios en red $[\widehat{M}_N]_{\lambda \mid x F}$.

Por lo tanto, la fila de la matriz de los f circuitos obteniendo en la etapa (3b) es convertido en una matriz diagonal $[B]_{F \times F}$ con los elementos dispuestos en la diagonal principal. A continuación, se realiza la operación de multiplicación entre la matriz de los movimientos unitarios y matriz cuadrada, obteniendo la matriz de movimientos unitarios en red del mecanismo.

$$[\widehat{M}_N]_{3.1 \times 5} = [[\widehat{M}_N]_{3 \times 5} \cdot \text{diag} \{[B_M]_{3 \times 5}\}]_{3.1 \times 5}$$

$$[\widehat{M}_N]_{3 \times 5} = \begin{bmatrix} -1 & -1 & -1 & -1 & 1 \\ 0 & -b_y & -c_y & -e_y & 0 \\ 0 & b_x & c_x & e_x & -d_x \end{bmatrix}_{3 \times 5} \quad \text{Ec.4.91.}$$

Etapa 5b. Ahora se adaptarán las leyes de Kirchhoff para la cinemática, esta etapa consiste en la aplicación de la ley de los circuitos resultando en el sistema de ecuaciones de la cinemática del mecanismo. En la ecuación 4.92 se muestra la separación de las velocidades angulares con respecto a la matriz de movimientos con el fin de despejar las variables a hallar.

$$[\widehat{M}_N]_{3 \times 5} \quad \{\phi\}_{5 \times 1} \quad \{\vec{0}\}_{3 \times 1}$$

$$\begin{bmatrix} -1 & -1 & -1 & -1 & 1 \\ 0 & -b_y & -c_y & -e_y & 0 \\ 0 & b_x & c_x & e_x & -d_x \end{bmatrix} \begin{Bmatrix} W_a \\ W_b \\ W_c \\ W_e \\ W_d \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}_{3 \times 1}$$

Ec.4.92.

La consistencia del sistema de ecuaciones depende del puesto m de la matriz $[\widehat{M}_N]_{3 \times 5}$ la forma más práctica de obtener del puesto es empleando las herramientas computacionales.

Para realizar el escalonamiento (operaciones entre las filas de la matriz) y en consecuencia identificar posibles ecuaciones redundantes. Por lo tanto puesto de la matriz permite determinar el número de variables primarias que deben ser impuestas al sistema.

En este caso la matriz $[\widehat{M}_N]_{3 \times 5}$ es llevada a su forma escalonada por filas, usando net en Matlab, resultando en la matriz $[\widehat{M}_{NESC}]_{3 \times 5}$

$$[\widehat{M}_{NESC}]_{3 \times 5} = \begin{bmatrix} 1 & 0 & 0 & A & D \\ 0 & 1 & 0 & B & E \\ 0 & 0 & 1 & C & F \end{bmatrix}_{3 \times 5}$$

Ec.4.93.

En donde

$$A = \frac{b_x c_y - b_y c_x - b_x e_y + b_y e_x + c_x e_y - c_y e_x}{b_x c_y - b_y c_x}$$

$$B = \frac{-(c_x e_y - c_y e_x)}{b_x c_y - b_y c_x}$$

$$C = \frac{b_x e_y - c_y e_x}{b_x c_y - b_y c_x}$$

$$D = \frac{(b_x c_y - b_y)(x + b_y d_x - c_y d_x)}{b_x c_y - b_y c_x}$$

$$E = \frac{-c_y d_x}{b_x c_y - b_y c_x}$$

$$F = \frac{b_y d_x}{b_x c_y - b_y c_x}$$

Recordando que el puesto de una matriz es el menor número de filas (o columnas) linealmente independientes, es posible observar que la matriz $[\widehat{M}_{NEsc}]_{3 \times 5}$ posee las 3 filas independientes consecuentemente al puesto de $[\widehat{M}_N]_{3 \times 5}$ es $m = 3$.

Etapa 6a. En esta etapa se halla el grado de restricción líquido C_N , o sea, hay redundancia en el sistema una vez que el puesto de $[\widehat{M}_N]_{3 \times 5}$ es $m = 3$, se aplica lo siguiente.

$$C_N = \lambda - m = 3 - 3 = 0$$

Ec.4.94.

Como se puede observar en la matriz escalonada de la etapa anterior (5b), este sistema no contiene redundancias por no tener resultado nulo en ninguna fila de la matriz. Esto se debe a que en la etapa (4a) la orden del sistema es $\lambda = 3$ para la construcción de los heligiros.

Etapa 6b. El sistema de ecuaciones escalonada (sin redundancias) resulta en la siguiente expresión:

$$\begin{matrix}
 [\hat{M}_{NESC}]_{m \times F} & \{\phi\}_{F \times 1} & \{\bar{0}\}_{m \times 1} \\
 \\
 \begin{bmatrix} 1 & 0 & 0 & A & D \\ 0 & 1 & 0 & B & E \\ 0 & 0 & 1 & C & F \end{bmatrix}_{3 \times 5} & \begin{Bmatrix} W_a \\ W_b \\ W_c \\ W_e \\ W_d \end{Bmatrix}_{5 \times 1} & = & \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}_{3 \times 1} & \text{Ec.4.95.}
 \end{matrix}$$

Etapa 7a. Esta etapa consiste en determinar las variables primarias del problema. Por tanto es necesario hallar el grado de libertad líquida del mecanismo.

$$F_N = F - m = 5 - 3 = 2 \quad \text{Ec.4.96.}$$

Por lo tanto 2 movimientos unitarios son necesarios para describir el comportamiento de los otros movimientos unitarios del problema.

Generalmente son seleccionados como variables primarias aquellas relacionadas con los actuadores que en este mecanismo de 5 barras pueden ser los acoples a y d (entrada de movimiento) y (salida de movimientos). En este caso las variables primarias seleccionadas con W_a y W_d , conociendo así la velocidad angular de la salida a determinadas velocidades angulares de entrada.

Etapa 7b. De esta forma el sistema de ecuaciones es dividido colocando de esta forma las velocidades angulares conocidas (en la parte superior) y las velocidades angulares que se quieren encontrar (en la parte inferior) luego se despejan las incógnitas como puede observarse en las ecuaciones 4.97 y 4.98:

$$[[\widehat{M}_{NS}]_{3 \times 3} \vdots [\widehat{M}_{NP}]_{3 \times 2}] \begin{Bmatrix} \{\phi_s\} & 3 \times 1 \\ \dots & \dots \\ \{\phi_p\} & 2 \times 1 \end{Bmatrix} = \{\vec{0}\}_{3 \times 1} \quad \text{Ec.4.97.}$$

$$\begin{bmatrix} 1 & 0 & A & 1 & D \\ 1 & 0 & B & 0 & E \\ 0 & 1 & C & 0 & F \end{bmatrix}_{3 \times 5} \begin{Bmatrix} W_b \\ W_c \\ W_e \\ \dots \\ W_a \\ W_d \end{Bmatrix}_{5 \times 1} = \begin{Bmatrix} 0 \\ 0 \\ 0 \end{Bmatrix}_{3 \times 1}$$

$$[\widehat{M}_{NS}]_{3 \times 3} \{\phi_s\}_{3 \times 1} = [\widehat{M}_{NP}]_{3 \times 2} \{\phi_p\}_{2 \times 1} \quad \text{Ec.4.98.}$$

$$\begin{bmatrix} 0 & 0 & A \\ 1 & 0 & B \\ 0 & 1 & C \end{bmatrix}_{3 \times 3} \begin{Bmatrix} W_b \\ W_c \\ W_e \end{Bmatrix}_{3 \times 1} = - \begin{bmatrix} 1 & D \\ 0 & E \\ 0 & F \end{bmatrix} \begin{Bmatrix} W_a \\ W_d \end{Bmatrix}$$

Etapla 8a. Despejando el vector de las variables secundarias el sistema de ecuaciones se torna

$$\{\phi_s\}_{3 \times 1} = -[\widehat{M}_{NS}]_{3 \times 3}^{-1} [\widehat{M}_{NP}]_{3 \times 2} \{\phi_p\}_{2 \times 1} \quad \text{Ec.4.99}$$

El cálculo de la matriz inversa $[\widehat{M}_{NS}]_{3 \times 3}^{-1}$ es:

$$\begin{Bmatrix} W_b \\ W_c \\ W_e \end{Bmatrix}_{3 \times 1} = - \begin{bmatrix} G & -1 & 0 \\ H & 0 & -1 \\ 1 & 0 & 0 \end{bmatrix}_{3 \times 3} \begin{bmatrix} 1 & D \\ 0 & E \\ 0 & F \end{bmatrix}_{3 \times 2} \begin{Bmatrix} W_a \\ W_d \end{Bmatrix}$$

En donde

$$G = \frac{-(c_x e_y - c_y e_x)}{b_x c_y - b_y c_x - b_x e_y + b_y e_x + c_x e_y - c_y e_x}$$

$$H = \frac{b_x e_y + b_y e_x}{b_x c_y - b_y c_x - b_x e_y + b_y e_x + c_x e_y - c_y e_x}$$

$$I = \frac{-(b_x c_y + b_y c_x)}{b_x c_y - b_y c_x - b_x e_y + b_y e_x + c_x e_y - c_y e_x}$$

Realizando las operaciones necesarias se obtiene la solución del sistema de ecuaciones de la cinemática con las variables secundarias en función de W_a y W_d .

Etapa 9a. La última etapa se refiere al pos procesamiento de las magnitudes obtenidas en la solución aplicados a los heligiros. El estado cinemático del mecanismo de 5 barras es obtenido analíticamente reemplazando los valores de las velocidades angulares secundarias en la ecuación (5).

Estática del mecanismo sin pierna (mecanismo de 5 barras)

Etapa 2a. En esta etapa se debe determinar las acciones externas, internalizadas y reunir las características de los acoples necesarios para la formación de las helifuerzas y la construcción del grafo de acciones 6a.

En el mecanismo de 5 barras, existen $e = 5$ acoples de tipo par rotativo, que tiene como característica $C_p = 2$ restricciones unitarias de fuerza (R_x, R_y) en el espacio plano ($\lambda = 3$)

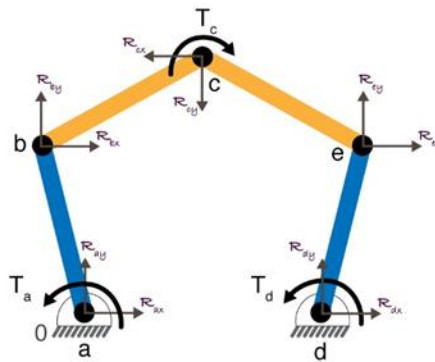
Se consideran $e_a = 3$ acoples activos: dos torques T_{in} en la dirección O_z siendo ejercido por fuentes motrices en las barras 1 y 4; y una resistencia al torque T_{out} , en la misma dirección,

representando una carga en las barras 2 y 3. Cada torque externo corresponde a $C_a = 1$ restricción unitaria y el mecanismo debe ser super restringido para realizar el análisis estático.

Las acciones externas son substituidas por acciones internas en el mecanismo. Los torques de entrada T_{in} y de salida T_{out} son substituidos por restricciones de torque T_a en el acople (a), T_d en el acople (d) y T_c en el acople de salida (c).

Es posible verificar las acciones existentes en los acoples del mecanismo.

Ilustración 33. Acciones en los acoples del mecanismo.



En el mecanismo existen restricciones paralelas en 3 direcciones coordenadas O_x, O_y, O_z . Por lo tanto existen 3 vectores de dirección unitaria \vec{S}^A para a formación de las helifuerzas de acuerdo con cada acople.

$$\vec{S}_x^A = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \vec{S}_y^A = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \vec{S}_z^A = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad \text{Ec.4.100.}$$

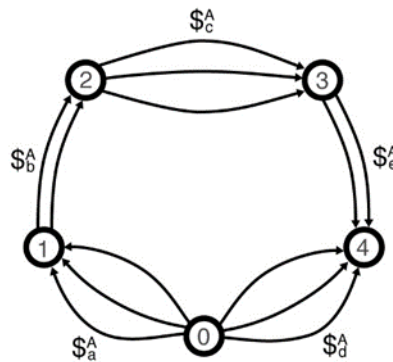
Los vectores de dirección unitaria x e y indican la dirección de las restricciones unitarias de los pares rotativos (a, b, c, d, e), en cuanto al vector z indica la dirección de las restricciones unitarias de los acoples activos (Tina, Tind, Toutc).

Los vectores de posición \vec{S}_o de las helifuerzas son idénticos a los heligros.

El paso de las helifuerzas de los pares rotativos es nulo ($h_{Rx,Ry}^A = 0$), entonces las restricciones consideradas son solamente de fuerza. El paso de los helifuerzas de los acoples activos es infinito ($h_{Tz}^A = \infty$), entonces las restricciones consideradas son solamente de momento.

Etapa 3a. Una vez que la geometría, la topología y las actuaciones externas de mecanismo son conocidas se forma el grafo de acciones GA . Cada una de las $e = 5$ aristas de Gc son sustituidas por $C_p = 2$ aristas en paralelo, representando las acciones pasivas. En las aristas a, d y c son adicionadas $C_a = 1$ arista en cada una, que son las acciones internalizadas T_a, T_d, T_c , respectivamente.

Ilustración 34. Grafo de acciones del mecanismo de 5 barras.



Calculando el grado de restricción total C del mecanismo se tiene

$$C = \sum_{i=1}^e C_{pi} + \sum_{j=1}^{ea} C_{aj} = (2 + 2 + 2 + 2 + 2) + (1 + 1 + 1) = 13 \quad \text{Ec.4.101.}$$

Como puede ser verificado en el grafo de acciones:

Etapa 3b. Considerada una de las más importantes del análisis estático, esta etapa es una conversión de las informaciones topológicas contenidas en el grado GA para la forma matricial, resultando en la matriz de los f cortes $[Q]_{KXC}$. Para esto es necesario identificar los f cortes.

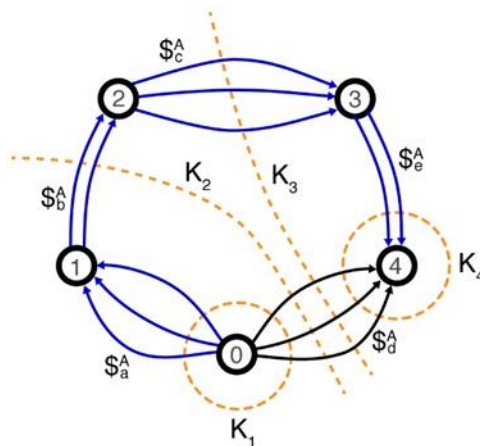
Primero se determina el número de f cortes k de la siguiente forma:

$$K = n - 1 = 5 - 1 = 4 \quad \text{Ec.4.102.}$$

Por lo tanto, se debe determinar cada uno de los $K = 4$ f cortes a través de la solución de un árbol generador.

Basándose en el árbol generador seleccionado en la etapa 3b de la cinemática, las aristas a, b, c y

Ilustración 35. f cortes K_1, K_2, K_3 y K_4 en el grafo de acciones del mecanismo.



Se obtiene la matriz de f cortes del mecanismo.

$$[Q_A]_{KXC} = [Q_A]_{4X13}$$

$$= \begin{bmatrix} a_{rx} & a_{ry} & b_{rx} & b_{ry} & c_{rx} & c_{ry} & e_{rx} & e_{ry} & d_{rx} & d_{ry} & aT_{in} & dT_{in} & CT_{out} \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \begin{matrix} \text{Corte 1} \\ \text{Corte 2} \\ \text{Corte 3} \\ \text{Corte 4} \end{matrix} \quad Ec. 4.103.$$

Las filas representan los f cortes y las columnas representan las restricciones unitarias de cada acople y también las acciones internalizadas.

Etap 4a. Dadas las características geométricas de los acoples, se debe construir las helifuerzas. De acuerdo con el paso serán usados en los casos particulares. Para $h^A = 0$ y para $h^A = \infty$. Las helifuerzas son dadas a continuación:

$$\$_{aRx}^A = \begin{pmatrix} \vec{S}_{oa} \times \vec{S}_x^A \\ \dots \\ \vec{S}_x^A \end{pmatrix} \quad R_{ax} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ R_{ax} \\ 0 \\ 0 \end{bmatrix}$$

$$\$_{aRy}^A = \begin{bmatrix} \vec{S}_{oa} \times \vec{S}_y^A \\ \dots \\ \vec{S}_y^A \end{bmatrix} \quad R_{ay} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ R_{ay} \\ 0 \end{bmatrix}$$

$$\$_{bRx}^A = \begin{bmatrix} \vec{S}_{ob} \times \vec{S}_x^A \\ \dots \\ \vec{S}_x^A \end{bmatrix} \quad R_{bx} = \begin{bmatrix} 0 \\ 0 \\ -b_y R_{bx} \\ \dots \\ R_{bx} \\ 0 \\ 0 \end{bmatrix}$$

$$S_{bRy}^A = \begin{bmatrix} \vec{S}_{ob} \times \vec{S}_y^A \\ \dots \\ \vec{S}_y^A \end{bmatrix} \quad R_{by} = \begin{bmatrix} 0 \\ 0 \\ b_x R_{by} \\ \dots \\ 0 \\ R_{by} \\ 0 \end{bmatrix}$$

$$S_{cRx}^A = \begin{bmatrix} \vec{S}_{oc} \times \vec{S}_x^A \\ \dots \\ \vec{S}_y^A \end{bmatrix} \quad R_{cx} = \begin{bmatrix} 0 \\ 0 \\ -c_y R_{cx} \\ \dots \\ R_{cx} \\ 0 \\ 0 \end{bmatrix}$$

$$S_{cRy}^A = \begin{bmatrix} \vec{S}_{oc} \times \vec{S}_y^A \\ \dots \\ \vec{S}_y^A \end{bmatrix} \quad R_{cy} = \begin{bmatrix} 0 \\ 0 \\ -c_x R_{cy} \\ \dots \\ 0 \\ R_{cy} \\ 0 \end{bmatrix}$$

$$S_{eRx}^A = \begin{bmatrix} \vec{S}_{oe} \times \vec{S}_x^A \\ \dots \\ \vec{S}_x^A \end{bmatrix} \quad R_{ex} = \begin{bmatrix} 0 \\ 0 \\ -e_y R_{ex} \\ \dots \\ R_{ex} \\ 0 \\ 0 \end{bmatrix}$$

$$S_{eRy}^A = \begin{bmatrix} \vec{S}_{oe} \times \vec{S}_y^A \\ \dots \\ \vec{S}_y^A \end{bmatrix} \quad R_{ey} = \begin{bmatrix} 0 \\ 0 \\ e_x R_{ey} \\ \dots \\ 0 \\ R_{ex} \\ 0 \end{bmatrix}$$

$$S_{eRx}^A = \begin{bmatrix} \vec{S}_{od} \times \vec{S}_x^A \\ \dots \\ \vec{S}_x^A \end{bmatrix} \quad R_{dx} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \dots \\ R_{dx} \\ 0 \\ 0 \end{bmatrix}$$

$${}^A dRy = \begin{bmatrix} \vec{S}_{od} x \vec{S}_y^A \\ \dots \\ \vec{S}_y^A \end{bmatrix} \quad R_{dy} = \begin{bmatrix} 0 \\ 0 \\ d_x R_{dy} \\ \dots \\ 0 \\ R_{dy} \\ 0 \end{bmatrix}$$

$${}^A aTin = \begin{bmatrix} \vec{S}_z^A \\ \dots \\ \vec{0} \end{bmatrix} \quad T_{ain} = \begin{bmatrix} 0 \\ 0 \\ T_{ain} \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^A dTin = \begin{bmatrix} \vec{S}_z^A \\ \dots \\ \vec{0} \end{bmatrix} \quad T_{din} = \begin{bmatrix} 0 \\ 0 \\ T_{din} \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$${}^A cTout = \begin{bmatrix} \vec{S}_z^A \\ \dots \\ \vec{0} \end{bmatrix} \quad T_{cout} = \begin{bmatrix} 0 \\ 0 \\ T_{cout} \\ \dots \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Ec.4.104.

Observando las helifuerzas se percibe que las coordenadas P^*, Q^*, N son nulas, evidenciando que las acciones existentes en el mecanismo están restringidas las coordenadas R^*, L, M del espacio del plano ($\lambda = 3$).

Por lo tanto, las helifuerzas son:

$$\begin{matrix} \$a_{Rx}^A = \begin{pmatrix} 0 \\ \dots \\ R_{ax} \\ 0 \end{pmatrix} & \$a_{aRy}^A = \begin{pmatrix} 0 \\ \dots \\ 0 \\ R_{ay} \end{pmatrix} & \$b_{bRx}^A = \begin{pmatrix} -b_y R_{bx} \\ \dots \\ R_{bx} \\ 0 \end{pmatrix} & \$b_{bRy}^A = \begin{pmatrix} b_x R_{by} \\ \dots \\ 0 \\ R_{by} \end{pmatrix} & \$c_{cRx}^A = \begin{pmatrix} -c_y R_{cx} \\ \dots \\ R_{cx} \\ 0 \end{pmatrix} \end{matrix}$$

$$\begin{matrix} \$c_{cRy}^A = \begin{pmatrix} c_x R_{cy} \\ \dots \\ 0 \\ R_{cy} \end{pmatrix} & \$e_{eRx}^A = \begin{pmatrix} -e_y R_{ex} \\ \dots \\ R_{ex} \\ 0 \end{pmatrix} & \$e_{eRy}^A = \begin{pmatrix} e_x R_{ey} \\ \dots \\ 0 \\ R_{ey} \end{pmatrix} & \$d_{dRx}^A = \begin{pmatrix} 0 \\ \dots \\ R_{dx} \\ 0 \end{pmatrix} & \$d_{dRy}^A = \begin{pmatrix} d_x R_{dy} \\ \dots \\ 0 \\ R_{dy} \end{pmatrix} \end{matrix}$$

$$\begin{matrix} \$a_{aTin}^A = \begin{pmatrix} T_{ain} \\ \dots \\ 0 \\ 0 \end{pmatrix} & \$d_{dTin}^A = \begin{pmatrix} T_{din} \\ \dots \\ 0 \\ 0 \end{pmatrix} & \$c_{cTout}^A = \begin{pmatrix} T_{cout} \\ \dots \\ 0 \\ 0 \end{pmatrix} \end{matrix} \quad \text{Ec.4.105.}$$

La línea punteada identifica que la parte superior es relativa a los momentos y la parte inferior corresponde a las fuerzas.

Etapla 4b. La matriz de las acciones $[A_D]_{\lambda x C}$ reúne todas las helifuerzas del mecanismo.

$$[A_D]_{3x13} =$$

$$\begin{bmatrix} 0 & 0 & -b_y R_{bx} & b_y R_{by} & -c_y R_{cx} & c_x R_{cy} & -e_y R_{ex} & e_x R_{ey} & 0 & d_x R_{dx} & T_{ain} & T_{din} & T_{cout} \\ R_{ax} & 0 & R_{bx} & 0 & R_{cx} & 0 & R_{ex} & 0 & R_{dx} & 1 & 0 & 0 & 0 \\ 0 & R_{ay} & 0 & R_{by} & 0 & R_{cy} & 0 & R_{ey} & 0 & R_{dy} & 0 & 0 & 0 \end{bmatrix}_{3 \times 13}$$

Ec.4.106.

Donde las columnas corresponden a las helifuerzas respectivamente.

Al separar las magnitudes de los heliguos, es posible obtener la matriz de acciones unitarias

$[\hat{A}_D]\lambda x C.$

$$[\hat{A}_D]_{3 \times 13} = \begin{bmatrix} 0 & 0 & -b_y & b_x & -c_y & c_x & -e_y & e_x & 0 & d_x & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}_{3 \times 13} \quad \text{Ec.4.107.}$$

Donde las columnas son las helifuerzas normalizadas.

El vector de las magnitudes de las acciones $\{\bar{\Psi}\}_{13 \times 1}$ es formado por:

$$\{\bar{\Psi}\}_{13 \times 1} = \begin{Bmatrix} R_{ax} \\ R_{ay} \\ R_{bx} \\ R_{by} \\ R_{cx} \\ R_{cy} \\ R_{ex} \\ R_{ey} \\ R_{dx} \\ R_{dy} \\ T_{ain} \\ T_{din} \\ T_{cout} \end{Bmatrix} \quad \text{Ec.4.108.}$$

Etapas 5a. En esta etapa, las informaciones geométricas y topológicas son reagrupadas nuevamente en la matriz de acciones unitarias en red $[\hat{A}_N]\lambda. KXC.$

Por tanto, cada fila de la matriz de los f cortes obtenida en la etapa (3_b) es convertida en una matriz cuadrada con los elementos dispuestos en la diagonal principal.

Utilizando las matrices diagonales, resulta la matriz de acciones unitarias en red $[\hat{A}_N]\lambda.KXC$ del mecanismo.

$$[\hat{A}_N]_{3.4 \times 13} = \begin{bmatrix} [\hat{A}_D]_{3 \times 13} \cdot \text{diag} \{[QA_1]_{1 \times 13}\} \\ [\hat{A}_D]_{3 \times 13} \cdot \text{diag} \{[QA_2]_{1 \times 13}\} \\ [\hat{A}_D]_{3 \times 13} \cdot \text{diag} \{[QA_3]_{1 \times 13}\} \\ [\hat{A}_D]_{3 \times 13} \cdot \text{diag} \{[QA_4]_{1 \times 13}\} \end{bmatrix}$$

$$[\hat{A}_N]_{12 \times 13} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & d_x & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & -b_y & b_x & 0 & 0 & 0 & 0 & 0 & d_x & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -c_y & c_x & 0 & 0 & 0 & d_x & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -e_y & e_x & 0 & d_x & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}_{12 \times 13}$$

Ec.4.109.

Etapa 5b. Se adaptan las leyes de Kirchhoff para la estática, esta etapa consiste en la aplicación de la ley de los cortes resultando en el sistema de ecuaciones de la estática del mecanismo.

$$[\hat{A}_N]_{12 \times 13} \{\phi\}_{13 \times 1} = \{\vec{0}\}_{12 \times 1}$$

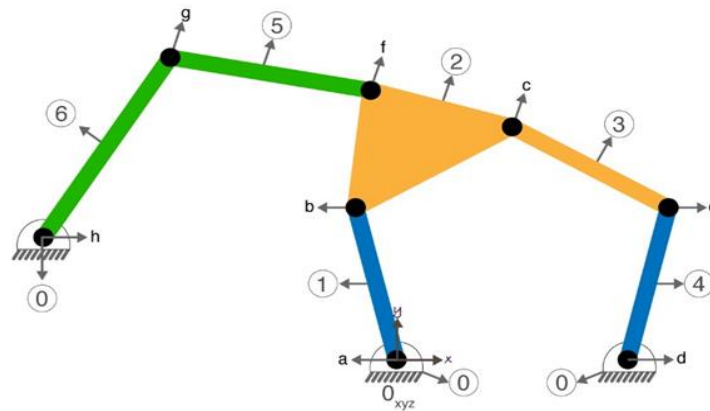
Ec.4.110.

b. Mecanismo con pierna

El mecanismo está compuesto por $n = 7$ cuerpos y por $e = 8$ acoples, el espacio de trabajo es $\lambda = 3$, ya que se trabaja en el plano.

Etapa 1a. En la etapa inicial se ilustra la representación esquemática del mecanismo. Es una versión simplificada del modelo físico conteniendo las informaciones de topología y geometría basado del sistema inercial de coordenadas O_{xyz}

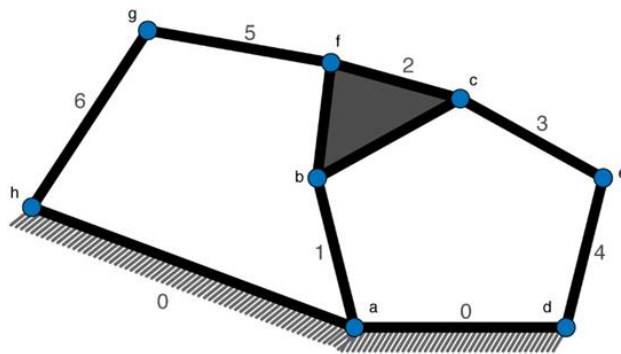
Ilustración 36. Representación esquemática del mecanismo con pierna.



En la ilustración 36 se puede observar el mecanismo hallado representado con los eslabones de colores azul y naranja, los eslabones de color verde hacen referencia a la pierna del paciente.

Etapa 1b. Se forma la red de acoples del mecanismo. Básicamente es una representación de la topología del mecanismo donde los cuerpos son polígonos cuyo número de vértices es el grado de conexión (número de acopes) en cada cuerpo.

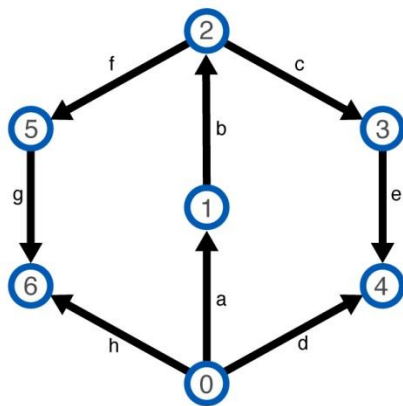
Ilustración 37. Red de acoples del mecanismo



En la ilustración 37 se muestra el esquema de acoples del mecanismo con la sujeción de la pierna.

Etapa 1c. Esta etapa trata de la construcción del grafo de acoples GC de la red, donde cada cuerpo es representado por una aresta, como lo muestra la siguiente figura.

Ilustración 35 Grafo de acoples GC de mecanismo.

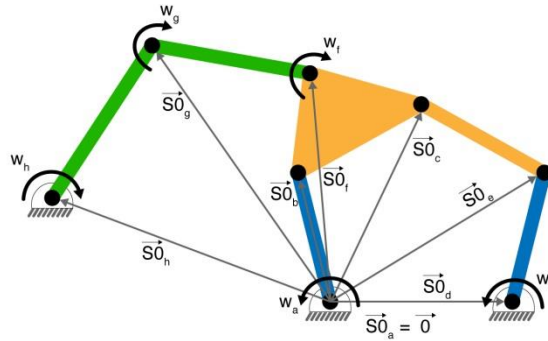


En la ilustración 35 se muestra la representación del mecanismo en grafo de acoples caracterizando el análisis hecho.

Cinemática del mecanismo con pierna.

Etapa 2a. Se hallan los vectores de posición y se determinan los movimientos del mecanismo.

Ilustración 38. Vectores de posición \vec{S}_o y los respectivos movimientos.



En la ilustración 35 se observan los diferentes heligios desde el origen del mecanismo.

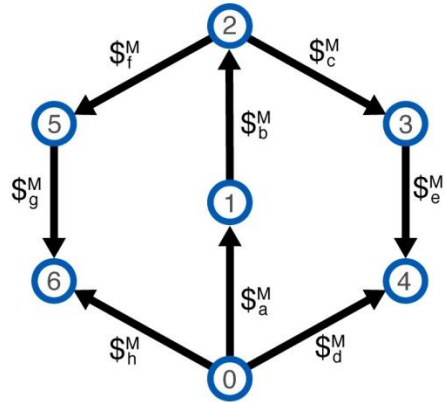
Los acoples de tipo par rotativo de $F = 1$. La posición de lo heligios es:

$$\begin{aligned}
 \vec{S}_{0a} &= \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} & \vec{S}_{0b} &= \begin{pmatrix} b_x \\ b_y \\ 0 \end{pmatrix} & \vec{S}_{0c} &= \begin{pmatrix} c_x \\ c_y \\ 0 \end{pmatrix} & \vec{S}_{0d} &= \begin{pmatrix} d_x \\ 0 \\ 0 \end{pmatrix} & \vec{S}_{0e} &= \begin{pmatrix} e_x \\ e_y \\ 0 \end{pmatrix} \\
 \vec{S}_{0f} &= \begin{pmatrix} f_x \\ f_y \\ 0 \end{pmatrix} & \vec{S}_{0g} &= \begin{pmatrix} g_x \\ g_y \\ 0 \end{pmatrix} & \vec{S}_{0h} &= \begin{pmatrix} h_x \\ h_y \\ 0 \end{pmatrix} & & & & & & \text{Ec.4.111.}
 \end{aligned}$$

Los Pasos De Los Heligios Son Nulos ($h^M = 0$)

Etapla 3a. Grafo De Movimientos GM.

Ilustración 39. Grafo De Movimientos GM Del Mecanismo.



En la ilustración 36 se muestran los movimientos del mecanismo con sus respectivas direcciones.

El Grado De Libertad Total Está Dado Por

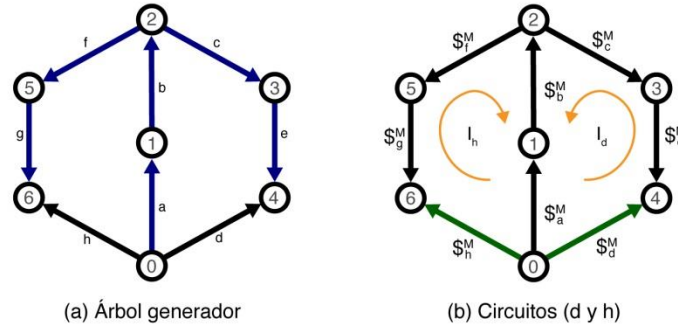
$$F = \sum_{i=1}^e fi = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 8 \quad \text{Ec.4.112.}$$

Etapas 3b. Se determina el número de f circuitos.

$$l = e - n + 1 - 7 + 1 = 2 \quad \text{Ec.4.113.}$$

Se obtiene el árbol generador como se ilustra en la ilustración 37.

Ilustración 40. Determinación de los f circuitos del mecanismo.



La matriz de los f circuitos es:

$$[B_M]_{|xf} = [B_M]_{2 \times 8} = \begin{bmatrix} a & b & c & d & e & f & g & h \\ -1 & -1 & -1 & -1 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & -1 & -1 & 1 \end{bmatrix}_{2 \times 8} \begin{matrix} |d \\ |h \end{matrix} \quad \text{Ec.4.114.}$$

Etapla 4a. Dadas las características geométricas de los acoples se deben construir los heligiros. Con o visto en las etapas anteriores todos los acoples de este mecanismo son el tipo par rotativo, indicando que los pasos de los heligiros son nulos.

Los heligiros son:

$$\begin{aligned} \$M_a^M &= \begin{pmatrix} w_a \\ 0 \\ 0 \end{pmatrix} & \$M_b^M &= \begin{pmatrix} w_b \\ b_y w_b \\ -b_x w_b \end{pmatrix} & \$M_c^M &= \begin{pmatrix} w_c \\ c_y w_c \\ -c_x w_c \end{pmatrix} & \$M_d^M &= \begin{pmatrix} w_d \\ 0 \\ -d_x w_d \end{pmatrix} \\ \$M_e^M &= \begin{pmatrix} w_e \\ e_y w_e \\ -e_x w_e \end{pmatrix} & \$M_f^M &= \begin{pmatrix} w_f \\ f_y w_f \\ -f_x w_f \end{pmatrix} & \$M_g^M &= \begin{pmatrix} w_g \\ g_y w_g \\ -g_x w_g \end{pmatrix} & \$M_h^M &= \begin{pmatrix} w_h \\ h_y w_h \\ -h_x w_h \end{pmatrix} \end{aligned} \quad \text{Ec.4.115.}$$

Etapa 4b. La matriz de los movimientos $[M_D]_{\lambda XF}$

$$[M_D]_{3 \times 8} = \begin{bmatrix} w_a & w_b & w_c & w_e & w_d & w_f & w_g & w_h \\ 0 & b_y w_b & c_y w_c & e_y w_e & 0 & f_y w_f & g_y w_g & h_y w_h \\ 0 & -b_x w_b & -c_x w_c & -e_x w_e & -d_x w_d & -f_x w_f & -g_x w_g & -h_x w_h \end{bmatrix}_{3 \times 8} \quad \text{Ec.4.116.}$$

La matriz de los movimientos unitarios $[\widehat{M}_D]_{\lambda XF}$

$$[\widehat{M}_D]_{3 \times 8} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & b_y & c_y & e_y & 0 & f_y & g_y & h_y \\ 0 & -b_x & -c_x & -e_x & -d_x & -f_x & -g_x & -h_x \end{bmatrix}_{3 \times 8} \quad \text{Ec.4.117.}$$

El vector de las magnitudes de los movimientos $\{\phi\}_{Fx1} =$

$$\{\phi\}_{8 \times 1} = \left\{ \begin{array}{c} w_a \\ w_b \\ w_c \\ w_e \\ w_d \\ w_f \\ w_g \\ w_h \end{array} \right\} \quad \text{Ec.4.118.}$$

Etapa 5a. Matriz de movimientos unitarios en red $[\widehat{M}_N]_{\lambda. |xF}$

$$[\widehat{M}_N]_{3.2 \times 8} = \begin{bmatrix} [\widehat{M}_N]_{3 \times 8} \cdot \text{diag} \{ [BM_1]_{1 \times 8} \} \\ [\widehat{M}_N]_{3 \times 8} \cdot \text{diag} \{ [BM_2]_{1 \times 8} \} \end{bmatrix}$$

$$[\widehat{M}_N]_{6 \times 8} = \begin{bmatrix} -1 & -1 & -1 & -1 & 1 & 0 & 0 & 0 \\ 0 & -b_y & -c_y & -e_y & 0 & 0 & 0 & 0 \\ 0 & b_x & c_x & e_x & -d_x & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & -1 & -1 & 1 \\ 0 & -b_y & 0 & 0 & 0 & -f_y & -g_y & h_y \\ 0 & b_x & 0 & 0 & 0 & f_x & g_x & -h_x \end{bmatrix} \quad \text{Ec.4.119.}$$

Etapa 5b. Se aplican las leyes de Kirchhoff resultando el sistema de ecuaciones de cinemática del mecanismo.

$$[\widehat{M}_N]_{6 \times 8} \{ \phi \}_{8 \times 1} = \{ \vec{0} \}_{6 \times 1}$$

Se escalona la matriz resultando $[\widehat{M}_{NEsc}]_{6 \times 8}$. Ninguna de las filas de la matriz es nula por lo tanto $m=6$.

Etapa 6a. Grado de restricción líquido C_N

$$C_N = \lambda \cdot | - m = 3.2 - 6 = 6 - 6 = 0 \quad \text{Ec.4.120.}$$

Etapa 6b. Sistema de ecuaciones escalonada.

$$[\widehat{M}_{NEsc}]_{m \times F} \{ \phi \}_{F \times 1} = \{ \vec{0} \}_{m \times 1}$$

Etapa 7a. Grado de libertad líquido.

$$F_N = F - m = 8 - 6 = 2 \quad \text{Ec.4.121.}$$

Etapa 7b. Sistema de ecuaciones particionado.

$$[[\widehat{M}_{NS}]_{6 \times 6} \quad : \quad [\widehat{M}_{NP}]_{6 \times 2}] \quad \begin{Bmatrix} \{\phi_s\}_{6 \times 1} \\ \{\phi\}_{2 \times 1} \end{Bmatrix} = \{\bar{0}\}_{6 \times 1} \quad \text{Ec.4.122.}$$

$$[\widehat{M}_{NS}]_{6 \times 6} \quad \{\phi_s\}_{6 \times 1} = -[\widehat{M}_{NP}]_{6 \times 2} \quad \{\phi_P\}_{2 \times 1} \quad \text{Ec.4.123.}$$

Etapa 8a. Se despeja el vector de variables secundarias.

$$\{\phi_s\}_{6 \times 1} = [\widehat{M}_{NS}]_{6 \times 6} \quad [\widehat{M}_{NP}]_{6 \times 2} \quad \{\phi_P\}_{2 \times 1} \quad \text{Ec.4.124.}$$

Se dejan las variables secundarias en función de w_a y w_d .

Etapa 9a. se hallan los heligiros.

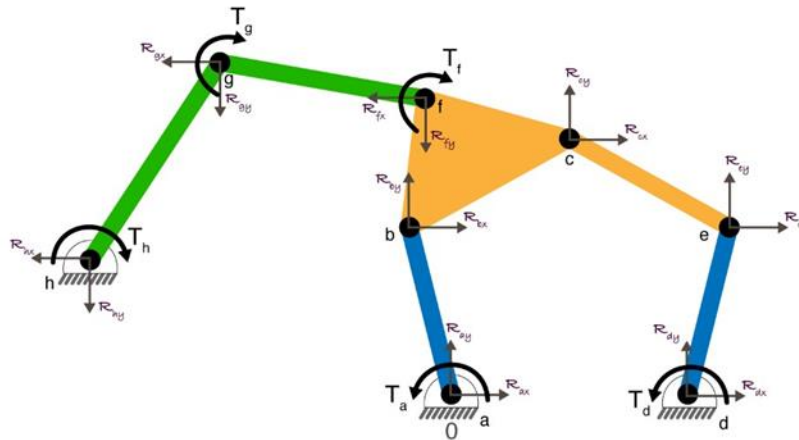
Estática del mecanismo con pierna.

Etapa 2a. Acciones en los acoples del mecanismo $C_p = 2$ $e_a = 5$ $C_a = 1$

Figura. Acciones en los acoples del mecanismo.

Etapa 3a. Grafo de acciones del mecanismo.

Ilustración 41. Grafo de acciones del mecanismo



Se calcula el grado de restricción total C

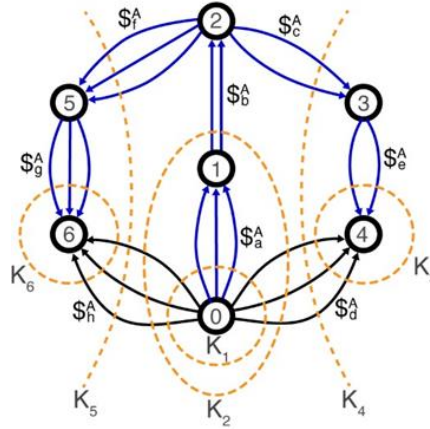
$$= \sum_{i=1}^e C_{pi} + \sum_{j=1}^{e_a} C_{aj} = (2 + 2 + 2 + 2 + 2 + 2 + 2 + 2) + (1 + 1 + 1 + 1 + 1) = 21 \quad \text{Ec.4.125.}$$

Etapa 3b. Determinar el número de f cortes k.

$$K = n - 1 = 7 - 1 = 6$$

Ec.4.126.

Ilustración 42. f cortes K_1, K_2, K_3, K_4, K_5 y K_6 en el grafo de acciones del mecanismo.



Se obtiene la matriz de f cortes.

$$[Q_A]_{K \times C} = [Q_A]_{6 \times 21}$$

$$= \begin{bmatrix} a_{rx} & a_{ry} & b_{rx} & b_{ry} & c_{rx} & c_{ry} & e_{rx} & e_{ry} & d_{rx} & d_{ry} & f_{rx} & f_{ry} & g_{rx} & g_{ry} & h_{rx} & h_{ry} & T_{ain} & T_{din} & T_{eout} & T_{gout} & T_{hout} \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}_{6 \times 21}$$

corte 1
 corte 2
 = corte 3
 corte 4
 corte 5
 corte 6

Ec.4.127.

Etapla 4a. $h^A = 0$ y $h^A = \infty$ las helifuerzas son:

$$\begin{aligned}
\$_{aRx}^A &= \begin{pmatrix} 0 \\ R_{ax} \\ 0 \end{pmatrix} & \$_{aRy}^A &= \begin{pmatrix} 0 \\ 0 \\ R_{ay} \end{pmatrix} & \$_{bRx}^A &= \begin{pmatrix} -b_y R_{bx} \\ R_{bx} \\ 0 \end{pmatrix} & \$_{bRy}^A &= \begin{pmatrix} b_x R_{by} \\ 0 \\ R_{by} \end{pmatrix} \\
\$_{cRx}^A &= \begin{pmatrix} -c_y R_{cx} \\ R_{cx} \\ 0 \end{pmatrix} & \$_{cRy}^A &= \begin{pmatrix} c_x R_{cy} \\ 0 \\ R_{cy} \end{pmatrix} & \$_{eRx}^A &= \begin{pmatrix} -e_y R_{ex} \\ R_{ex} \\ 0 \end{pmatrix} & \$_{eRy}^A &= \begin{pmatrix} e_x R_{ey} \\ 0 \\ R_{ey} \end{pmatrix} \\
\$_{dRx}^A &= \begin{pmatrix} 0 \\ R_{dx} \\ 0 \end{pmatrix} & \$_{dRy}^A &= \begin{pmatrix} d_x R_{dy} \\ 0 \\ R_{dy} \end{pmatrix} & \$_{fRx}^A &= \begin{pmatrix} -f_y R_{fx} \\ R_{fx} \\ 0 \end{pmatrix} & \$_{fRy}^A &= \begin{pmatrix} f_x R_{fy} \\ 0 \\ R_{fy} \end{pmatrix} \\
\$_{gRx}^A &= \begin{pmatrix} -g_y R_{gx} \\ R_{gx} \\ 0 \end{pmatrix} & \$_{gRy}^A &= \begin{pmatrix} g_x R_{gy} \\ 0 \\ R_{gy} \end{pmatrix} & \$_{hRx}^A &= \begin{pmatrix} -h_y R_{hx} \\ R_{hx} \\ 0 \end{pmatrix} & \$_{hRy}^A &= \begin{pmatrix} h_x R_{hy} \\ 0 \\ R_{hy} \end{pmatrix} \\
\$_{aTin}^A &= \begin{pmatrix} T_{ain} \\ 0 \\ 0 \end{pmatrix} & \$_{dTin}^A &= \begin{pmatrix} T_{din} \\ 0 \\ 0 \end{pmatrix} & \$_{fTout}^A &= \begin{pmatrix} T_{fout} \\ 0 \\ 0 \end{pmatrix} & \$_{gTout}^A &= \begin{pmatrix} T_{gout} \\ 0 \\ 0 \end{pmatrix} & \$_{hTout}^A &= \begin{pmatrix} T_{hout} \\ 0 \\ 0 \end{pmatrix}
\end{aligned}$$

Ec.4.128.

Etapla 4b. La matriz de acciones $[A_D]_{\lambda x C}$

$$\begin{aligned}
& [A_D]_{3 \times 21} \\
& = \begin{bmatrix} 0 & 0 & -b_y R_{bx} & b_x R_{by} & -c_y R_{cx} & c_x R_{cy} & -e_y R_{ey} & e_x R_{ex} & 0 & d_x R_{dy} & -f_y R_{fx} & f_x R_{fy} & -g_y R_{gx} & g_x R_{gy} & -h_y R_{hx} & h_x R_{hy} & T_{ain} & T_{din} & T_{eout} & T_{gout} & T_{hout} \\ R_{ax} & 0 & R_{bx} & 0 & R_{cx} & 0 & R_{ex} & 0 & R_{dx} & 1 & R_{fx} & 0 & R_{gx} & 0 & R_{hx} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & R_{ay} & 0 & R_{by} & 0 & R_{cy} & 0 & R_{ey} & 0 & R_{dy} & 0 & R_{fy} & 0 & R_{gy} & 0 & R_{hy} & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{3 \times 21}
\end{aligned}$$

La matriz de acciones unitaria

$$[\hat{A}_D]_{3 \times 21} = \begin{bmatrix} 0 & 0 & -b_y & b_x & -c_y & c_x & -e_y & e_x & 0 & d_x & -f_y & f_x & -g_y & g_x & -h_y & h_x & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{3 \times 21} \quad \text{Ec.4.129.}$$

El vector de las magnitudes de las acciones:

$$\{\bar{\Psi}\}_{cx1}$$

$$\{\bar{\Psi}\}_{21 \times 1} = \{R_{ax} R_{ay} R_{bx} R_{by} R_{cx} R_{cy} R_{ex} R_{ey} R_{dx} R_{dy} R_{fx} R_{fy} R_{gx} R_{gy} R_{hx} R_{hy} T_{ain} T_{din} T_{eout} T_{gout} T_{hout}\}$$

Etapla 5a. Matriz de acciones unitarias en $R_{ed}[\hat{A}N] \lambda. KXC$

$$[\hat{A}N]_{3.6 \times 12} = \begin{bmatrix} [\hat{A}_D]_{3 \times 21} \cdot \text{diag} \{[QA_1]_{1 \times 21}\} \\ [\hat{A}_D]_{3 \times 21} \cdot \text{diag} \{[QA_2]_{1 \times 21}\} \\ [\hat{A}_D]_{3 \times 21} \cdot \text{diag} \{[QA_3]_{1 \times 21}\} \\ [\hat{A}_D]_{3 \times 21} \cdot \text{diag} \{[QA_4]_{1 \times 21}\} \\ [\hat{A}_D]_{3 \times 21} \cdot \text{diag} \{[QA_5]_{1 \times 21}\} \\ [\hat{A}_D]_{3 \times 21} \cdot \text{diag} \{[QA_6]_{1 \times 21}\} \end{bmatrix}$$

Etapla 5b. Se adaptan las leyes de Kirchhoff para hallar el sistema de ecuaciones de la estática.

$$[\hat{A}_N]_{18 \times 21} \{\phi\}_{21 \times 1} = \{\vec{0}\}_{18 \times 1}$$

Se halla la matriz escalonada $[\hat{A}_{NEsc}]_{18 \times 21} \quad a = 18$

Etapa 6a. Grado de libertad líquido F_N

$$F_N = \lambda.K - a = 3.6 - 18 = 18 - 18 = 0 \quad \text{Ec.4.130.}$$

Etapa 6b. Sistema de ecuaciones escalonado.

$$[\hat{A}_{NESC}]_{axc} \{\phi\}_{cx1} = \{\vec{0}\}_{ax1}$$

Etapa 7a. Grado de restricción líquido C_N

$$C_N = c - a = 21 - 18 = 3 \quad \text{Ec.4.131.}$$

Se toman las variables primaria $T_{fout}, T_{gout}, T_{hout}$

Etapa 7b. Sistema de ecuaciones resultante.

$$[[\hat{A}_{NS}]_{18 \times 18} \quad \vdots \quad [\hat{A}_{NP}]_{18 \times 3}] \begin{Bmatrix} \{\vec{\Psi}_S\}_{18 \times 1} \\ \{\vec{\Psi}_P\}_{3 \times 1} \end{Bmatrix} = \{\vec{0}\}_{18 \times 1}$$

$$[\hat{A}_N]_{18 \times 18} \{\overline{\Psi}_S\}_{18 \times 1} = -[\hat{A}_{NP}]_{18 \times 1} \{\overline{\Psi}_P\}_{3 \times 1}$$

Etapas 8a. Se resuelve el sistema de ecuaciones.

Anexo 24

Etapas 9a. Se obtiene el estado estático del mecanismo.

Anexo 25

3.2. Control de posición de las barras del mecanismo que permita establecer arcos de movilidad y tiempos que se deben aplicar para cada uno de los movimientos necesarios en la rehabilitación de lesión de rodilla.

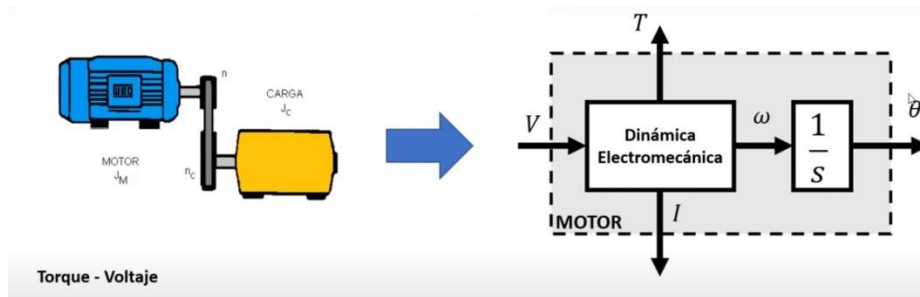
Se tomó la decisión de usar estos actuadores teniendo en cuenta los cálculos obtenidos por medio del Método de Davies. Los actuadores usados para posicionar las barras del mecanismo son dos motores DC, estos fueron escogidos gracias a la construcción necesaria de los mismos para poder llevar a cabo el trabajo requerido. Su construcción es la de un motor DC cuyo eje se acopla a un mecanismo de transmisión sin fin corona el cual ofrece la posibilidad de una vez sea desconectado de su fuente de alimentación conserve su posición angular, esto debido a la fricción de esta construcción.

Modelamiento de Motor DC

Para conseguir el modelo matemático de los motores de corriente directa que se utilizan en este proyecto se tuvieron en cuenta diferentes modelos analíticos y experimentales para conseguir el modelo, puesto que los motores no contaban con características de fabricante se procedió a determinar las características de cada uno de ellos, por medio de experimentos y medidas con instrumentos teniendo en cuenta que no existió la posibilidad de acceder a laboratorios se usaron instrumentos electrónicos como la tarjeta Arduino, multímetro, circuitos comparadores entre otros.

Después de comparar la respuesta de cada uno de los modelos se utilizó el de mejor comportamiento para hallar un controlador clásico PID que permitiera el correcto funcionamiento de los motores. Se realiza la aclaración que aunque los dos motores de corriente directa son iguales en su construcción, se analizaron de manera separada obteniendo pequeñas diferencias en sus características que hacían necesarias el modelamiento de manera diferente. De la siguiente manera se desarrollan los análisis para encontrar el modelo matemático.

Ilustración 44. Modelo matemático del motor DC



Ecuaciones eléctricas

Utilizando la segunda ley de Kirchhoff

Circuito de excitación:

$$V_f = R_f i_f + L_f \frac{di_f}{dt} \quad \text{Ec. 5.1.}$$

Donde:

V_f : Voltaje de excitación.

R_f : Resistencia de excitación.

i_f : Corriente de excitación.

L_f : Inductancia de excitación

Circuito de la armadura:

$$V_a = e_a + L_a \frac{di_a}{dt} + R_a i_a \quad \text{Ec. 5.2.}$$

Donde e_a es una tensión generada que resulta cuando dos conductores de la armadura se mueven a través de un flujo de campo establecido por la corriente de campo i_f

$$e_a = K_e \omega_m \quad \text{Ec. 5.3.}$$

Se asume que existe una relación proporcional K , entre el voltaje inducido en la armadura y la velocidad angular del eje.

Ecuación mecánica

Utilizando la segunda ley de Newton par el movimiento rotatorio.

$$\sum M = M_e - M_B - M_c = J_T \frac{dwm}{dt} \quad \text{Ec. 5.4.}$$

Sabiendo que el par viscoso tiene $M_B = B_T W$ donde B_T es la constante viscosa.

$$M_e = J_T \frac{dwm}{dt} + B_T Wm + M_c \quad \text{Ec. 5.5.}$$

$$J_T \frac{dwm}{dt} = M_e - B_T Wm - M_c \quad \text{Ec. 5.6.}$$

Donde:

J_T = Inercia del motor y de la carga referida al eje del motor.

B_T = Constante viscosa del motor y de la carga referida al eje del motor.

W_m = Velocidad angular.

M_e = Par electromagnético.

M_B = Par viscoso.

M_c = Par mecánico de la carga

Se supone la siguiente relación electromagnética que establece que el torque mecánico es proporcional a K a la corriente eléctrica.

$$M_e = K_m i_a \quad \text{Ec.5.7.}$$

Ecuaciones de modelo analítico

Se puede considerar que si un motor de corriente directa de excitación independiente trabaja con el valor de la corriente del campo constante, $I_f = V_f / R_f$ y también la carga es lineal, entonces el sistema puede ser considerado como un sistema lineal.

Las ecuaciones son:

$$V_a = K W_m + L_a \frac{di_a}{dt} + R_a i_a \quad \text{Ec.5.8.}$$

Ecuación de la armadura

$$M_e = K_m i_a \quad \text{Ec.5.9.}$$

Ecuación electromagnética del par

$$M_e = J_T \frac{dW_R}{dt} + B_T W_R + M_c \quad \text{Ec. 5.10.}$$

Ecuación mecánica

Dicho modelo tiene dos variables de entrada o excitaciones, V_a y M_c , por lo que debe utilizarse el principio de superposición para combinar las respuestas del estado estable y transitorio en el funcionamiento del sistema para ambas entradas.

Obtención del modelo analítico en diagrama de bloques.

El comportamiento transitorio puede ser analizado transformando las ecuaciones del modelo del sistema para condiciones iniciales igual a cero. Si aplicamos la transformada de Laplace.

$$V_a (S) = K_e W (S) + S L_a i_a (S) + R_a i_a (S) \quad \text{Ec.5.11.}$$

$$M_e (S) = K_m i_a (S) = S J_T W(S) + B_r W(S) + M_c (S) \quad \text{Ec.5.12.}$$

Despejado $i_a (S)$ de (5.11)

$$i_a (S) = \frac{V_a (S) - K_e W (S)}{L_a S + R_a} = \frac{[V_a (S) - K_e W (S)]}{R_a (T_a S + 1)}$$

$$= [V_a (S) - K_e W (S)] \left(\frac{1}{R_a (T_a S + 1)} \right) \quad \text{Ec.5.13.}$$

Donde la constante de tiempo del circuito de armadura es $T_a = \frac{L_a}{R_a}$ Ec.5.14.

Despejando $W (S)$ de (12)

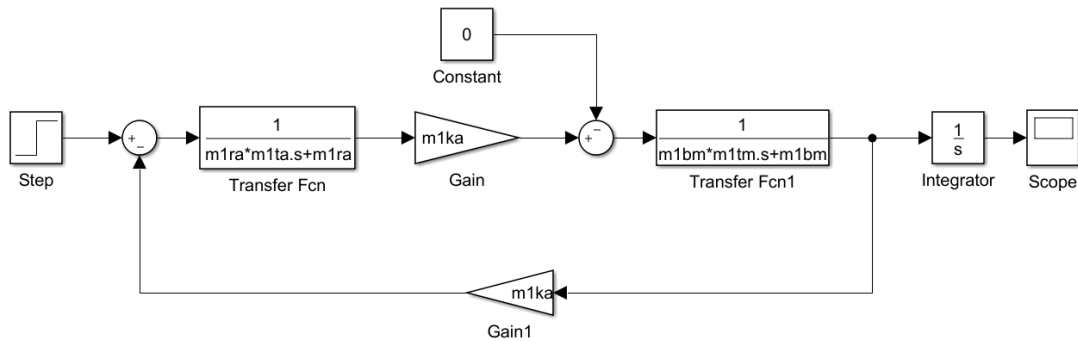
$$W (S) = \frac{M_e (S) - M_c (S)}{J_T S + B_r} = \frac{[M_e (S) - M_c (S)]}{B_T (T_m S + 1)}$$

$$= [M_e (S) - M_c W (S)] \left(\frac{1}{B_T (T_m S + 1)} \right) \quad \text{Ec. 5.15.}$$

Donde lo constante de tiempo mecánica del motor y de la carga es $T_m = \frac{J_T}{B_T}$ Ec. 5.16.

Diagrama de bloque de las ecuaciones anteriores.

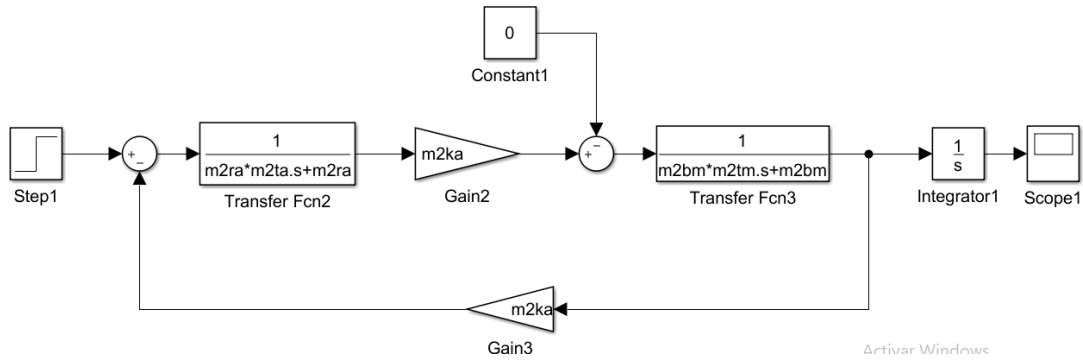
Ilustración 45. Modelo analítico caso 1 Motor DC 1



Está obteniendo como salida controlada la velocidad angular, pero lo que se necesita en este caso es el control de la posición, por lo tanto lo que se debe agregar al diagrama de bloques anterior es un integrador a la salida de velocidad angular.

Modelado analítico de Motor DC Caso 2

Ilustración 46. Modelo analítico caso 1 Motor DC 2



Tomando del caso 1 desde la ecuación (5.7) se aplica la transformada de Laplace a las ecuaciones (5.2), (5.3), (5.6) y (5.7) obteniendo

$$L_a S i_a (S) = V_a (S) - R_a i_a (S) - e_a (S) \quad \text{Ec.5.17.}$$

$$e_a (S) = K_e W_m (S) \quad \text{Ec.5.18.}$$

$$J_T S W (S) = M_e (S) - B_r W_m (S) \quad \text{Ec.5.19.}$$

$$M_e (S) = K_m i_a (S) \quad \text{Ec.5.20.}$$

Aquí se desprecia $M_c (S)$ representando el modelo con carga en el eje igual a cero.

Se sustituye (17) y (19) en (16)

$$L_a S = \frac{M_e (S)}{K_m} = V_a (S) - R_a \frac{M_e (S)}{K_m} - K_e W_m (S)$$

$$V_a (S) = \frac{(R_a + L_a S) M_e (S)}{K_m} + K_e W (S) \quad \text{Ec. 5.21.}$$

De (18) se obtiene la velocidad angular $W (S)$

$$W_m(S) = \frac{M_e(S)}{J_s + B_T} \quad \text{Ec. 5.22.}$$

Se sustituye (21) en (20)

$$V_a(S) = \frac{(R_a + L_a S)M_e(S)}{K_m} + K_e \frac{M_e(S)}{J_s + B_T}$$

$$V_a(S) = \frac{(R_a + L_a S) (J_T S + B_T) K_e K_m}{K_m (J_T S + B_T)} M_e(S) \quad \text{Ec. 5.23.}$$

De esta forma se pueden obtener las funciones de transferencia que relacionen las salidas con las entradas siguientes:

Torque – Voltaje Ec.5.24.

$$\frac{M_e(S)}{V_a S} = \frac{K_m (J_T S + B_T)}{L_a J_T S^2 + (R_a J_T + L_a B_T) S + R_a B_T + K_e K_m}$$

Fem inducido – voltaje

Ec.5.25.

$$\frac{e_a(S)}{V_a S} = \frac{K_e K_m}{L_a J_T S^2 + (R_a J_T + L_a B_T) S + R_a B_T + K_e K_m}$$

Corriente de armadura – voltaje

Ec.5.26.

$$\frac{i_a(S)}{V_a S} = \frac{J_T S + B_T}{L_a J_T S^2 + (R_a J_T + L_a B_T) S + R_a B_T + K_e K_m}$$

Velocidad angular – voltaje Ec.5.27.

$$\frac{W_m(S)}{V_a S} = \frac{K_m}{L_a J_T S^2 + (R_a J_T + L_a B_T)S + R_a B_T + K_e K_m}$$

Posición angular – voltaje Ec.5.28.

$$\frac{\theta(S)}{V_a S} = \frac{K_m}{S (L_a J_T S^2 + (R_a J_T + L_a B_T)S + R_a B_T + K_e K_m)}$$

Representación en espacios de estado

Definir estados

$$\begin{aligned} X_1 &= W_m & \dot{X}_1 &= \dot{W}_m \\ X_2 &= i_a & \dot{X}_2 &= \dot{i}_m \end{aligned}$$

Se sustituye en las ecuaciones diferenciales.

$$\dot{X}_1 = -\frac{B_T}{J_T} X_1 + \frac{K_M}{J_T} X_2 \quad \text{Ec. 5.29.}$$

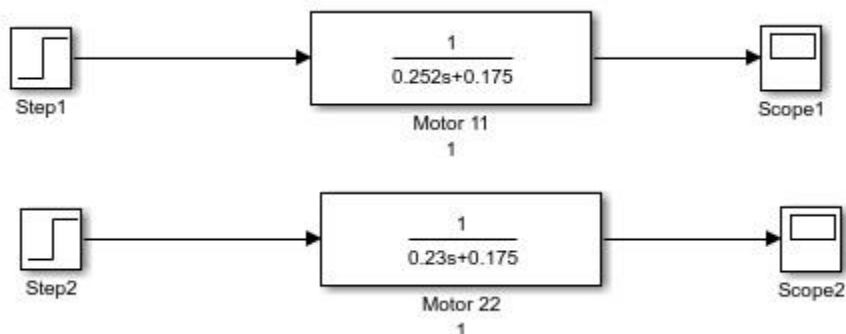
$$\dot{X}_2 = -\frac{R_a}{L_a} X_2 - \frac{K_e}{L_a} X_1 + \frac{1}{L_a} V_a \quad \text{Ec. 5.30.}$$

$$\begin{bmatrix} \dot{X}_1 \\ \dot{X}_2 \end{bmatrix} = \begin{bmatrix} -\frac{B_T}{J_T} & \frac{K_M}{J_T} \\ -\frac{K_e}{L_a} & -\frac{R_a}{L_a} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L_a} \end{bmatrix} V_a$$

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \quad \text{Ec.5.31.}$$

Se realiza el diagrama de bloques del modelo analítico, este sin torque de perturbación.

Ilustración 47. Modelo analítico caso 2 sin perturbación de carga



Modelado analítico de Motor DC Caso 3

Simplificación de las ecuaciones del Motor DC por el método de eliminación de la constante eléctrica del motor.

Las ecuaciones del motor DC pueden simplificarse teniendo en cuenta que la constante eléctrica del motor se desprecia, lo que traduce en despreciar el factor de la inductancia L_a en la ecuación eléctrica (5.2)

$$V_a = e_a + R_a i_a \quad \text{Ec.5.32.}$$

Se toma (5.5) desprecian M_c

$$M_e = J_T \frac{dW_m}{dt} + B_T W_m \quad \text{Ec. 5.33.}$$

Se reemplaza (5.3) y (5.32) en (5.30)

$$V_a = K_e W_m + R_a \left(\frac{J_T}{K_m} \right) \frac{dW_m}{dt} + R_a \left(\frac{B_T}{K_m} \right) W_m$$
$$V_a = \frac{R_a J_T}{K_m} \frac{dW_m}{dt} + \left(\frac{R_a B_T}{K_m} + K_e \right) W_m \quad \text{Ec. 5.34.}$$

La ecuación de primer orden (5.33) se le aplica la transformada de Laplace y la función de transferencia queda:

$$\frac{W_m(S)}{V_a(S)} = \frac{K_m}{R_a J_T S + R_a B_T + K_e K_m} \quad \text{Ec. 5.35.}$$

Se puede escribir la función de transferencia de la siguiente forma:

$$G(S) = \frac{W_m(S)}{V_a(S)} = \frac{K_m}{S + P_a} \quad \text{Ec. 5.36.}$$

La ganancia a bajas frecuencias del motor es.

$$G(0) = \frac{W_m(0)}{V_a(0)} = \frac{K_m}{R_a B_T + K_e K_m} \quad \text{Ec. 5.37.}$$

Relacionando (35) y (36) se obtiene:

$$K_m = P_a * G(0) \quad \text{Ec.5.38.}$$

El polo tiene un valor de

$$S - P_a = \frac{R_a B_T + K_e K_m}{R_a J_T} \quad \text{Ec. 5.39.}$$

Por lo tanto, el diagrama de bloques estaría de la siguiente forma

Simplificación de las ecuaciones del motor DC por el método de eliminación del polo no dominante.

La función de transferencia de la velocidad angular se puede expresar como:

$$G(S) = \frac{K_m}{(S + |P_1|) + (S + |P_2|)} \quad \text{Ec. 5.40.}$$

Donde los polos P_1 y P_2 se pueden hallar por la relación.

$$P_{1,2} = -\frac{1}{2} \left(\frac{1}{T_a} + \frac{1}{T_m} \right) \pm \frac{1}{2} \sqrt{\left(\frac{1}{T_a} + \frac{1}{T_m} \right)^2 - 4 \frac{K_m K_e}{J_T L_a}} \quad \text{Ec. 5.41.}$$

La ganancia a bajas frecuencias del motor es

$$G(0) = \frac{\dot{K}_m}{P_1 P_2} \quad \text{Ec. 5.42.}$$

Normalicen $P_1 \ll P_2$ por lo tanto se desprecia P_1

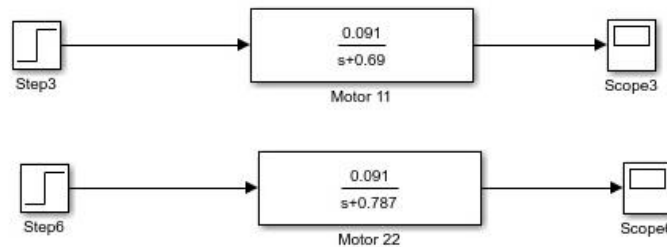
$$G(S) = \frac{\dot{K}_m}{S + P_2} \quad \text{Ec. 5.43.}$$

Teniendo en cuenta que

$$\begin{aligned} & \dot{K}_m \\ &= \frac{K_m}{(J_T L_a |P_1|)} \end{aligned} \quad \text{Ec. 5.44.}$$

En la ilustración 45 se muestran los valores del modelado de los motores DC descrito en las ecuaciones anteriores.

Ilustración 48. Modelo analítico caso 3

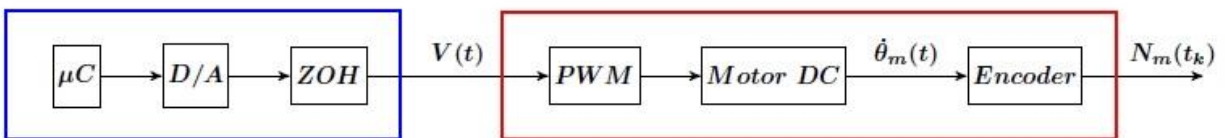


Modelado experimental de un motor DC Caso 1

Los motores DC suelen tener un comportamiento lineal en un rango predefinido de tensiones de entrada. Sin embargo a bajas velocidades angulares la fricción produce no linealidades, además que el motor sufre a su entrada ruidos debido al efecto de la carga y a la salida ruido de la medida de los sensores.

Otras no linealidades que se pueden presentar son debido a los errores del control digital, conversor A/D , conversor D/A , puente 4 y PWM , el ruido generado por estos es difícil de modelar.

Ilustración 49. Modelo Experimental Caso 1



Se modela un motor DC que incluya toda la electrónica que será utilizada en el control realimentado.

Para obtener la posición angular se representa $N_m(t_k)$ que representa el número de pulsos medido en un tiempo (t_k) en el eje.

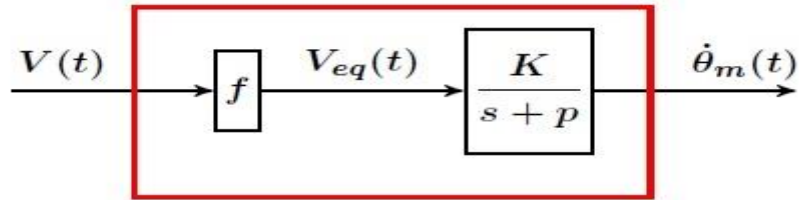
$$\theta_m(t_k) = \frac{2\pi N_m(t_k)}{q} \quad \text{Ec. 5.45.}$$

Donde q es el número de pulsos por vuelta (CPR, Cycles por Revolution) en el encoder incremental.

Se impondrá un modelo matemático que corresponda a un sistema lineal de primer orden en velocidad angular de tal manera que el objetivo se reduzca a dos parámetros del motor K y P donde K/P sea la ganancia a bajas frecuencias y P represente el único polo de un sistema de primer orden.

Se propone un modelado sencillo, que da lugar al modelo lineal deseado junto con un elemento no lineal de ajuste de la señal de entrada.

Ilustración 50. Modelo Experimental Caso 1 Simplificado



Modelo no lineal de un motor DC

Donde $V(t)$ representa la señal de entrada al motor y $V_{eq}(t) = fv(t)$ siendo la función no lineal resultante del modelado.

Obtención experimental del modelo simplificado (k, p) del motor DC.

En primer lugar, debe realizarse un conjunto de $P * Q$ experimentos: P repeticiones de Q experimentos con valores de entrada un escalón de diferente amplitud.

La velocidad angular de un motor DC se obtiene usando la primera aproximación de Euler de la derivada:

$$W_t(t_k) = \frac{2 \pi N_j(t_k) - N_j(t_k - 1)}{qT} \quad \text{Ec. 5.46.}$$

Donde T es el periodo de muestreo expresado en segundos, esta se medirá para diferentes amplitudes de voltaje.

$$VE = \{V_1, V_2, \dots, V_Q\} \quad \text{Ec. 5.47.}$$

El modelo que se propone consta de cuatro fases:

Primera fase: Obtención del valor medio de velocidades angulares del régimen permanente. Sobre el experimento promedio se obtendrá un valor medio de las últimas muestras en donde se garantice que el motor se encuentre en régimen permanente.

Si la discretización del tiempo es uniforme, T , y el escalón tuviese una duración de $T_F = (N_f - 1) T$ se escogen los valores de las últimas N muestras, esta media sería $W(\infty)$

$$\bar{W}(\infty) = \sum_{i=1}^N \bar{W} N_f - N + i \quad \text{Ec. 5.48.}$$

Se determina una cantidad Q de valores de la entrada, escalón $VE \{V_1, V_2, \dots, V_Q\}$, entonces se obtendrá un conjunto de valores medios de velocidad angular.

$$\{\bar{W}_1(\infty)\bar{W}_2(\infty), \dots, \bar{W}_Q(\infty)\} \quad \text{Ec.5.49.}$$

Si el motor se comportara linealmente los valores $\bar{W}_j(\infty)$, con $j = 1, 2, \dots, Q$ sería proporcionales tal que:

$$\frac{\bar{W}_j(\infty)}{\bar{W}_k(\infty)} = \frac{V_j}{V_k} \quad \text{Ec. 5.50.}$$

Para cualquier j y $k \in \{1, 2, \dots, Q\}$

Segunda fase: Se calcula el polo medio de varianza mínima.

Para el cálculo del polo de varianza mínima deberá estudiarse el comportamiento del motor en el régimen transitorio.

Esta fase se realiza para cada uno de los Q experimentos usando los valores $\bar{W}_j(\infty)$ obtenidos en la primera fase.

$$\{P_1, P_2, \dots, P_Q\} \quad \text{Ec.5.51.}$$

La expresión del polo que puede utilizarse para el transitorio de subida de una señal cuadrada es:

$$P S_j(k) = \frac{N_j(k) - N_j(k-1)}{T \frac{q W_j(\infty)}{2\pi} KT - N_j(k)} \quad \text{Ec. 5.52.}$$

La ecuación (5.51) se deduce de la discretización del polo del motor obtenido con una entrada escalón bajo condiciones nulas.

Al realizar un experimento con un micro controlador se obtiene una secuencia de valores discretos de la salida del motor de la siguiente forma:

$$P(t_k) = \frac{W_m(t_k)}{W_m(\infty) t_k - \theta(t_k)} \quad \text{Ec. 5.53.}$$

Donde $t_k = KT$ que es un instante de tiempo que depende de T que es el periodo de muestreo.

Se hace una aproximación de su valor utilizando la aproximación de Euler para la derivada, es decir

$$W_m(t_k) \approx \frac{\theta_m(t_k) - \theta_m(t_k - 1)}{t_k - t_k - 1} \quad \text{Ec. 5.54.}$$

Bajo condiciones iniciales nulas $K = 0$ no se puede tener velocidades angulares pero el experimento se toma con condiciones nulas.

$$W_m(t_o) = W_m(0) = 0$$

Si se supone que la posición angular viene dada por pulsos del encoder incremental entonces:

$$P(t_k) = \frac{N_m(t_k) - N_m(t_k - 1)}{(t_k - t_k - 1) \left(\frac{q W_m(\infty)}{2\pi} t_k - N_m(t_k) \right)} \quad \text{Ec. 5.55.}$$

Por otro lado la expresión del polo que puede utilizarse para el transitorio de bajada de una señal cuadrada es

$$pB_j = \frac{\frac{q W_j(\infty)}{2\pi}}{N_j(K_f) - N_j(K_o)} \quad \text{Ec. 5.56.}$$

Donde $K_f T$ es cualquier instante de tiempo en donde el motor se encuentra parado.

La ecuación (5.55) se deduce de la discretización del polo del motor obtenido con una señal de entrada cuadrada, retomando desde la ecuación (5.54).

El promedio temporal de la velocidad angular de toda la trayectoria de bajada, usando la aproximación de Euler para la derivada es:

$$w = \sum_{j=K_o+1}^{K_p} [(\theta(j) - \theta(j-1))] = \theta(K_p) - \theta(K_o) \quad \text{Ec. 5.57.}$$

Si $K_f \geq K_p$ con el motor parado, entonces

$$\frac{1}{P} = \frac{\theta(K_f) - \theta(K_o)}{W_m(\infty)} \quad \text{Ec. 5.58.}$$

Una vez calculados pS_j y pB_j se obtendrá un único valor P_j como una combinación lineal de los anteriores.

$$P_j = \alpha pS_j + (1 - \alpha)pB_j \quad \text{Ec. 5.59.}$$

Donde $\alpha \in \{0, 1\}$

Tercera fase: Cálculo de K y de la tensión de entrada equivalente.

Para calcular k en cada experimento se realiza con la fórmula.

$$K_j = \frac{P_j \bar{W}_j(\infty)}{V_j} \quad j \in \{1, 2, \dots, Q\} \quad \text{Ec. 5.60.}$$

Obteniendo el vector

$$\{K_1, K_2, \dots, K_Q\} \quad \text{Ec. 5.61.}$$

El problema que surge es que existe un modelo de motor para cada entrada (K_j, P_j)

Como se necesita un único modelo lineal junto con una corrección no lineal de la entrada del motor.

Para ello se selecciona, en primer lugar un polo común a cualquier entrada al motor, sea P este polo entonces

$$\bar{K}_j = \frac{P W_j(\infty)}{V_j} \quad \{j \in 1, 2, \dots, Q\} \quad \text{Ec. 5.62.}$$

Los valores de K_j será distintos para cada j obteniéndose el vector

$$\{K_1, K_2, \dots, K_Q\} \quad \text{Ec. 5.63.}$$

A partir de este vector puede obtenerse un valor K común a todas las entradas, como consecuencia el valor $\bar{W}_j(\infty)$ va a ser diferente.

$$\bar{W}_j(\infty) \neq \frac{K V_j}{K} \quad j \in \{1, 2, \dots, Q\} \quad \text{Ec. 5.64.}$$

Para corregir esto es suficiente obtener una entrada equivalente de valor:

$$V_{eqj} = \frac{\bar{K}_j V_j}{K} \quad j \in \{1, 2, \dots, Q\} \quad \text{Ec. 5.65.}$$

Por lo tanto el vector

$$\{V_{eqj1}, V_{eqj2}, \dots, V_{eqjQ}\} \quad \text{Ec. 5.66.}$$

Entonces

$$\frac{\bar{W}_j(\infty)}{V_{eqj}} = \frac{K}{P} \quad j \in \{1, 2, \dots, Q\} \quad \text{Ec. 5.67.}$$

Con esta nueva entrada de V_{eqj} se conservan las proporciones dadas por el mismo valor de P y K , V_{eqj} puede describirse de la forma:

$$V_{eqj} = \frac{K_j V_j}{P_j} = \frac{P}{K} \quad \text{Ec. 5.68.}$$

Cuarta fase: Identificación de la función no lineal del modelo del motor mediante la técnica de interpolación polinómica.

La parte no lineal del modelo del motor será una función no lineal f , tal que

$$V_{eq}(f) = f(V(t)) \quad \text{Ec.5.69.}$$

Se propone una función no lineal f de un polinomio de grado $N = 2Q$

$$f(x) = a_1X + a_2X^2 + \dots + aNX^N \quad \text{Ec.5.70.}$$

Donde a_i con $i = 1, 2, \dots, N$ son N constantes del modelo que deben ser identificados.

Si la no linealidad del motor es independiente del signo de su entrada el polinomio $f(x)$ será simétrico con respecto al origen, es decir, $f(-x) = -f(x)$, lo que permite reducir el grado del polinomio a $N - 1$, y el número de coeficientes a $N/2$ de tal forma que:

$$f(x) = a_1X + a_3X^3 + a_5X^5 + \dots + a_{2Q-1}X^{2Q-1} \quad \text{Ec.5.71.}$$

Se realiza el mismo método de interpolación con V_{eqj} con las mismas incógnitas

$$V_{eqj} = a_1V_j + a_2V_j^2 + \dots + aNV_j^N \quad \text{Ec.5.72.}$$

Y la misma condición de simetría

$$-V_{eqj} = -a_1V_j + a_2V_j^2 - a_3V_j^3 + \dots - a_{2Q-1}V_j^{2Q-1} + a_{2Q}V_j^{2Q} \quad \text{Ec.5.73.}$$

Restando y dividiendo por 2 las ecuaciones (73) y (74) se obtiene

$$V_{eqj} = a_1V_j + a_3V_j^3 + \dots + a_{2Q-1}V_j^{2Q-1} \quad \text{Ec.5.74.}$$

Así se obtendrá un sistema de Q ecuaciones lineales con Q incógnitas.

Este sistema de ecuaciones puede ser representado en forma matricial.

$$V_{eq} = V_a \quad \text{Ec.5.75.}$$

Donde

$$V_{eq} = \begin{bmatrix} V_{eq1} \\ V_{eq2} \\ \vdots \\ V_{eqQ} \end{bmatrix} \quad \text{Ec.5.76.}$$

$$a = \begin{bmatrix} a_1 \\ a_3 \\ \vdots \\ a_{2Q-1} \end{bmatrix} \quad \text{Ec.5.77.}$$

$$V = \begin{bmatrix} V_1 & V_1^3 & \dots & V_1^{2Q-1} \\ V_2 & V_2^3 & \dots & V_2^{2Q-1} \\ \vdots & \vdots & \ddots & \vdots \\ V_Q & V_Q^3 & \dots & V_Q^{2Q-1} \end{bmatrix} \quad \text{Ec.5.78.}$$

El vector de parámetros a se obtendría invirtiendo la matriz V .

$$a = V^{-1} V_{eq} \quad \text{Ec.5.79.}$$

Teniendo en cuenta la ecuación (66) se obtiene el valor de $\tilde{K}_j = K (a_1 + a_3 V_j^2 + \dots + a_{2Q-1} V_j^{2Q-1})$ Ec.5.80.

Esta expresión indica claramente que el valor de K depende de la entrada V_j . El valor de $K_{eq}(t)$ de la siguiente forma

$$K_{eq}(t) = K (a_1 + a_3 V^2(t) + \dots + a_{2Q-1} V_j^{2Q-1}(t)) \quad \text{Ec.5.81.}$$

De esta forma

$$V_{eq}(t) = \frac{K_{eq}(t)}{K} V(t) \quad \text{Ec.5.82.}$$

Toma de datos Motor DC

Los motores a usar tenían las características para poder realizar un modelamiento analítico de los motores DC, por lo tanto, se realizaron experimentos con los datos que se podían tomar y con los que se sabían del motor.

Para el caso del modelamiento experimental se tomaron los datos que se piden y se desarrollan las ecuaciones y procedimientos descritos en el modelamiento.

En esta parte se va a agregar la toma de datos que era parte de los anteriores procedimientos para generar una función de transferencia dependiendo de valores de entrada de voltaje y salida de velocidad angular, usando para esto la herramienta “ident” (System Identification) de matlab en cual es utilizado para crear modelos dinámicos lineales y no lineales a partir de datos medidos de entrada y salida, de esta forma poder obtener la función de transferencia que puede expresar la planta.

Toma de datos para el modelo analítico del motor DC Casos 1, 2, 3, 4

Se realiza la toma de medidas y valores de los motores DC, esto debido a que no es posible modelar un solo motor que sirva para los dos casos, debido a que así sean iguales no son idénticos y tienen diferencias pequeñas pero que pueden afectar el funcionamiento.

El único valor conocido es el voltaje de entrada del motor el cual es 12V, ya que estos motores son los utilizados para el movimiento de los vidrios en los carros.

Tabla 2. Características de R_a , L_a y V_a de los motores DC.

	Motor 1	Motor 2
R_a	1,7 Ω	1,5 Ω
L_a	51,88 uH	51,88 uH
V_a	12V	12V

Donde

R_a = Resistencia de armadura

L_a =Inductancia de armadura

V_a =Voltaje de armadura

Usando herramientas tales como multímetro, tarjetas de Arduino, entre otros componentes electrónicos se hizo posible estimar estos valores.

Para la R_a solo fue necesario el multímetro y medir la resistencia de los cables de los dos motores.

Para la L_a fue necesario realizar un montaje de circuito electrónico con la ayuda de una Arduino, uno para hallar le valor.

Tabla 3. Características de PWM , V_{IN} , E_B e I

PWM	Motor 1				Motor 2			
	V	V_{IN}	E_B	I	V	V_{IN}	E_B	I
255	12,53	12,30	0,23	0,69	12,53	12,30	0,23	0,81
200	10,4	11,45	1,05	0,67	10,4	11,45	1,05	0,74
150	8,45	10,55	2,1	0,62	8,45	10,55	2,1	0,68
128	7,58	10	2,42	0,62	7,58	10	2,42	0,68
100	6,5	8,85	2,35	0,58	6,5	8,85	2,35	0,65
64	5,09	6,15	1,06	0,54	5,09	6,15	1,06	0,60

Donde

PWM = En los pulsos del controlador al puente H .

V = Voltaje a la salida del puente H sin motor.

V_{IN} = Voltaje a la salida del puente H con motor.

E_b = fem inducida por el motor.

I = Corriente del motor.

En este caso se usa una tarjeta Arduino para generar el PWM que va al puente H el cual debido a esta modulación alimentará a diferentes voltajes los motores debido a que los motores cuentan con embobinados e imanes estos al ser accionados inducen una fem .

Para hallar la E_b se usa la siguiente fórmula:

$$E_b = |V - V_{IN}| \quad \text{Ec.5.83.}$$

Los demás valores son hallados con multímetro.

A partir de estos parámetros podemos encontrar la constante de tiempo eléctrica L_a con la ecuación (13)

Tabla 4. Constante de t_e

	Motor 1	Motor 2
L_a	0,031 ms	0,034 ms

Se tomaron los W_m de cada motor a diferentes pulsos de PWM , en este caso se utilizaron dos tarjetas de arduino, una para activar y enviar el PWM al puente H y la otra tarjeta para medir por medio de interrupciones los pulsos que envía el encoder incremental que mide la salida del *ese* del reductor del motor.

Tabla 5. Velocidades angulares

	Motor 1	Motor 2
PWA	W_m	W_m
255	27,9	27,085
200	25,32	24,78
150	22,987	22,76
128	21,15	21,38
100	18,49	18,905
64	12,735	12,33

Donde

PWM = Son los pulsos del controlador al puente H .

W_m = Velocidad angular.

Para obtener la velocidad angular en cada uno de los valores de PWM dados se utilizó la ecuación (45)

Tabla 6. Constante eléctrica y mecánica K_e y K_m

	Motor 1	Motor 2
PWA	$K_a = K_m$	$K_a = K_m$
255	0,0022	0,0084
200	0,041	0,042
150	0,091	0,092
128	0,114	0,113
100	0,127	0,124
64	0,083	0,085

Donde:

PWA = Son los pulsos del controlador al puente H .

K_a = Constante eléctrica

K_m = Constante mecánica

Se toma la consideración que $K_a = K_m$

De estos valores se puede observar que con $PWA = 255$ tiene un valor de voltaje mayor de los $12V$ que corresponden a la salida de la fuente de voltaje, estos valores al ser mayor que el voltaje recomendado para la entrada del motor DC y por lo tanto la fem inducida tiene valores erróneos y no se tienen en cuenta.

Para determinar las constantes se saca un promedio de los valores medibles, en donde:

$$K_a = (0,041 + 0,091 + 0,114 + 0,127 + 0,083) / 5$$

$$K_a = 0,091 = K_m \text{ Para motor 1}$$

$$K_a = (0,042 + 0,092 + 0,113 + 0,124 + 0,085) / 5$$

$$K_a = 0,091 = K_m \text{ Para motor 2}$$

Tabla 7. Características t_m , J_m y B_m

	Motor 1	Motor 2
t_m	0,033 mS	0,037 mS
J_m	$160,74 * 10^{-9} \text{ kgm}^2$	$204,26 * 10^{-9} \text{ kgm}^2$
B_m	4,87 mC	5,52 mC

Donde

$$t_m = (t_a * k_a) + t_a \quad \text{Ec.5.84.}$$

$$J_m = \frac{(t_m)(k_m)(k_a)}{R_a} \quad \text{Ec. 5.85.}$$

$$B_m = \frac{J_m}{t_m} \quad \text{Ec. 5.86.}$$

Con estos valores hallados se usan los modelos analíticos y se reemplazan para observar el comportamiento de estos modelos y analizar cuáles representando mejor forma los motores DC.

Toma de datos para el modelo experimental de motor DC Caso 1.

Para estos datos se toman las velocidades angulares tal como se hizo en los datos tomados en la toma de datos del modelo analítico.

Aparte de esto se usan las ecuaciones (51), (55) y (60) para hallar el polo de la función de transferencia.

Para hallar estos valores se usaron dos tarjetas arduino y tomando los datos con los tiempos estimados en el modelo experimental.

Tabla 8. Características modelo experimental.

PWA	Motor 1				Motor 2			
	W_m	pS	pB	p	W_m	pS	pB	p
255	27,9	280,4	43,5	161,9	27,025	288,2	42,40	135,3
200	25,32	4,81	39,64	22,22	24,78	4,49	38,79	21,87
150	22,987	2,55	35,99	19,27	22,76	2,71	35,63	19,17
128	21,15	2,12	33,11	17,62	21,38	2,59	33,47	18,03
100	18,49	2,20	28,95	15,57	18,905	1,57	29,6	15,58
64	12,735	3,30	19,94	11,62	12,33	3,21	19,3	11,25

Toma de datos para el uso de la herramienta Ident.

Debido a los errores que se pudieron cometer a la hora de medir las características del motor se hace necesario usar herramientas que permitan obtener un modelo de planta según datos de

entrada y salida que calcula una función de transferencia, intentando dar le comportamiento más cercano a los motores.

En este caso también se deben tomar los datos para cada motor debido a que pueden tener diferencias que afecten su comportamiento.

Tabla 9. Datos para la herramienta Ident

<i>PWA</i>	Motor 1		Motor 2	
	<i>V</i>	<i>M</i>	<i>V</i>	<i>M</i>
255	12,53	27,9	12,53	27,085
240	12,48	28,28	12,48	28,50
225	11,82	27,05	11,82	28,40
210	11,15	26,41	11,15	26,68
200	10,4	26,23	10,4	26,26
185	10,05	26,18	10,05	25,98
170	9,40	24,57	9,40	25,82
155	8,78	23,54	2,78	24,12
150	8,45	22,987	2,45	22,76
135	7,98	22,38	7,98	22,56
128	7,58	21,15	7,58	21,38
120	7,36	20,91	7,36	20,94
105	6,76	20,39	6,76	19,90
100	6,5	18,49	6,5	18,905
90	6,16	17,66	6,16	17,41
75	5,56	15,44	5,56	15,22
64	5,09	12,735	5,09	12,33
60	1,96	12,38	4,96	11,73
45	4,39	7,95	4,39	8,12
30	3,77	4,54	3,77	4,54
15	3,18	0	3,18	0
0	2,50	0	2,50	0

Donde

V = Voltaje de entrada

W = Velocidad angular

Estos datos serán ingresados la herramienta el voltaje V como entrada y la velocidad angular W como salida.

Toma de datos de la relación de reducción.

Los motores cuentan con un mecanismo de reducción llamado sin fin el cual está constituido de un tornillo sin fin y un engranaje los cuales reducen el número de vueltas que da el motor y aumenta el torque de salida.

Estas características no vienen especificadas en los motores por lo que se debe desarmar el motor y así poder medir el tornillo sin fin y la corona y determinar la reducción, esta se hace con un calibrador (pie de rey).

Largo del tornillo sin fin = $12,2mm$

Diámetro de la corona = $55,3 mm$

Altura del diente = $2,25 mm$

La ecuación que describe la altura del diente es

$$H = 2,167 (M) \quad \text{Ec.5.87.}$$

Por lo tanto,

$$M = \frac{2,25 mm}{2,167} \approx 1$$

Se toma el módulo $M = 1$

Ya que el paso del tornillo sin fin se calcula con la ecuación:

$$P = \pi M \quad \text{Ec.6.88.}$$

Entonces

$$P = \pi(1) = 3,1416$$

Teniendo en cuenta la ecuación del largo del tornillo que depende del número de dientes de la corona N .

$$L = P \left(4,5 + \frac{N}{50} \right) \quad \text{Ec. 5.89.}$$

De esta se despeja N

$$N = \left(\frac{L}{P} - 4,5 \right) 50 = 64,66 \approx 65$$

Al trabajar con un sin fin corona se puede tomar que el número de dientes de relación del tornillo es $n =$, por lo tanto

$$W_{IN} * n = W_{out} * N \quad (91) \quad \frac{W_{out}}{W_{IN}} = \frac{n}{N} \quad Ec. 5.90.$$

Por lo tanto la relación de reducción del eje el motor con el eje de salida está dando por

$$W_{out} = \frac{n}{N} W_{IN} \quad Ec. 5.91.$$

Entonces

$$W_{out} = \frac{1}{65} W_{IN} \Rightarrow W_{out} = 0,015 W_{IN}$$

Aplicación de los modelos de motor DC

Después de abordar las diferentes formas de modelar el motor DC tanto de forma analítica como experimental y tomar los datos que permitan caracterizar el motor para reemplazar estos valores en los modelos hallados para de esta forma poder realizar la simulación, observar de esta forma el comportamiento y ver cual modelo se asemeja más a la realidad en su comportamiento para de esta forma hallar un controlador que pueda hacer funcionar el motor de la forma requerida.

Para poder realizar este procedimiento se usa la herramienta Simulink de Matlab para observar el comportamiento de cada una de las funciones de transferencia y modelos realizados determinando de esta forma el modelo que más se aproxima para de esta forma sintetizar un controlador PID.

Una vez hallados los datos de las características del motor DC se reemplaza en el modelo hallado y se simula el comportamiento. En este caso se presenta el siguiente resultado.

Ilustración 51. Respuesta modelo analítico caso 1 del motor DC 1

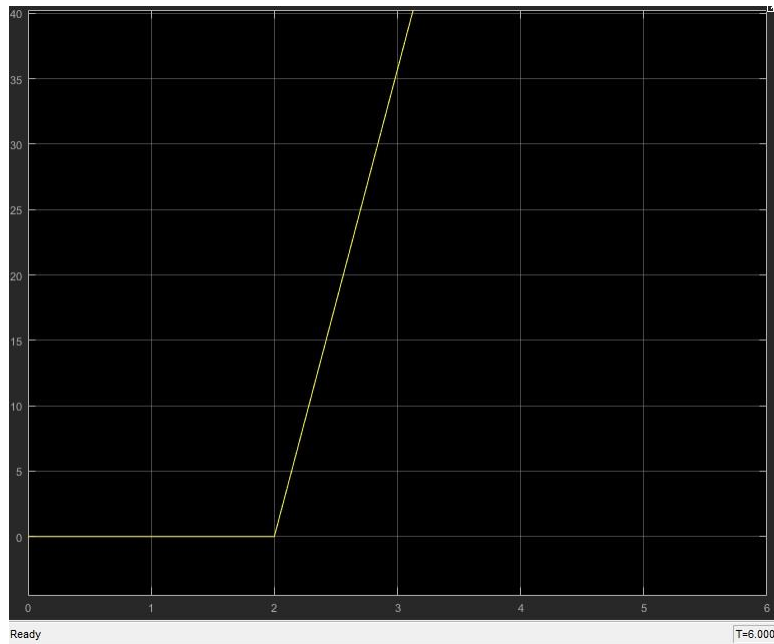
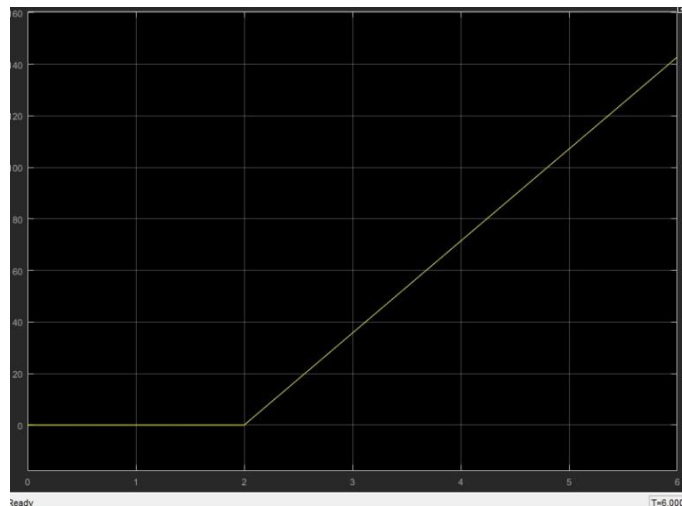


Ilustración 52. Respuesta modelo analítico caso 1 del motor DC 2



Se observa que el comportamiento del modelo no corresponde al del motor en el valor de salida ante una entrada de escalón en donde la entrada es voltaje y la salida es velocidad angular.

Debido a esto se descarta este modelado y se intenta con los demás casos que se han de analizar.

Modelo analítico de motor DC Caso 2

Al reemplazar los datos en el modelo del motor DC se observa que la constante que acompaña el término de segundo orden es muy pequeño por lo tanto este se desprecia y se observa el siguiente resultado:

Ilustración 53. Respuesta modelo analítico caso 2 del motor DC 1

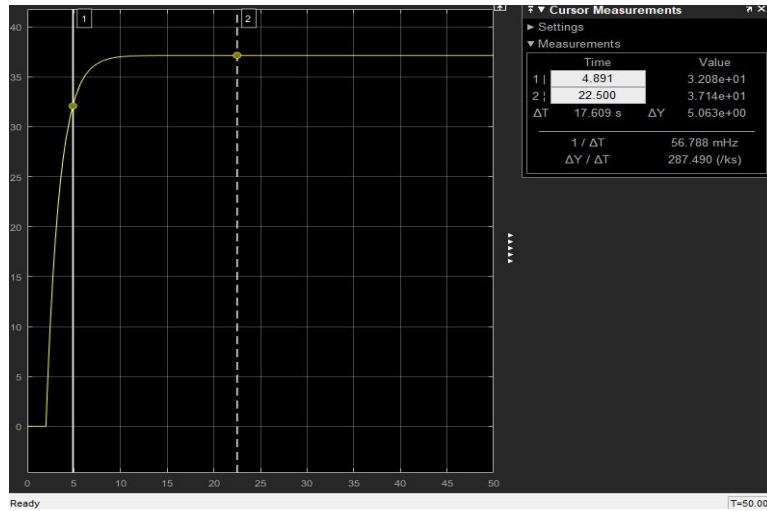
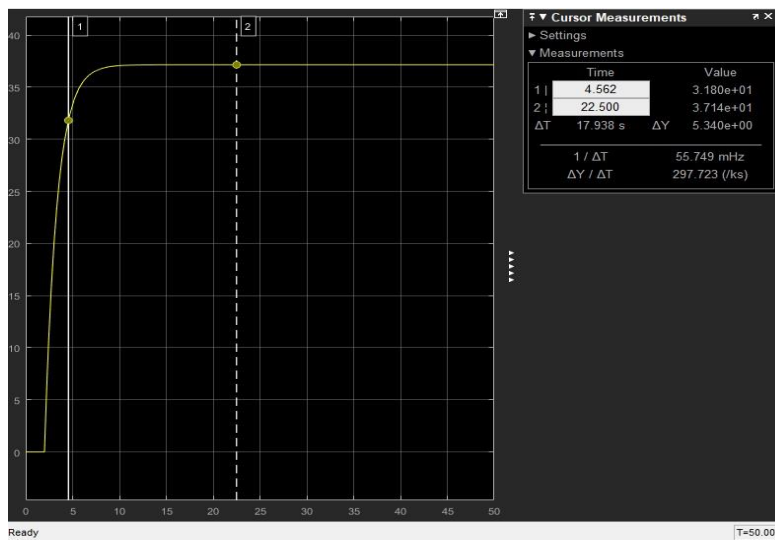


Ilustración 54. Respuesta modelo analítico caso 2 del motor DC 2



Se observa que el comportamiento del modelado es muy parecido al obtenido en el Caso 1 por lo tanto no corresponde al del motor en el valor de salida ante una entrada de escalón.

Debido a esto se descarta este modelado y se sigue intentando con los demás casos que se ha de analizar.

Modelo analítico del motor DC Caso 3

Este modelo es más simple que los anteriores, ya que, se halla un polo y con la constante mecánica se hace una función de transferencia de primer orden y se observa lo siguiente:

Ilustración 55. Respuesta modelo analítico caso 3 del motor DC 1

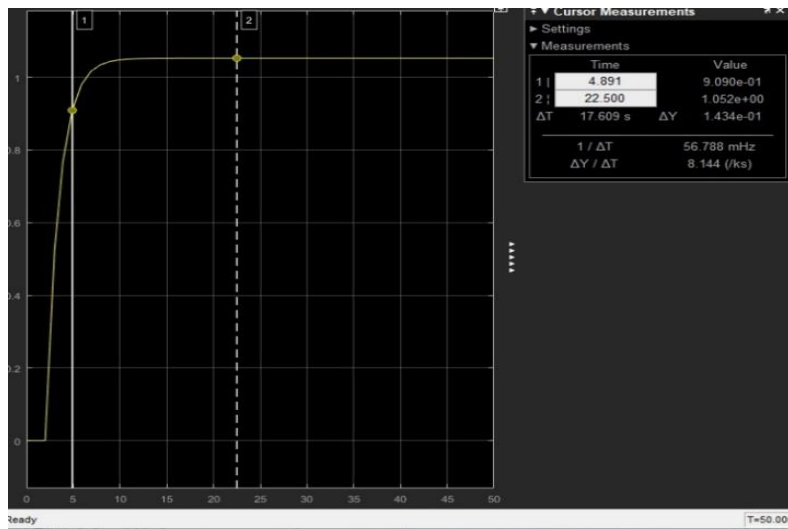
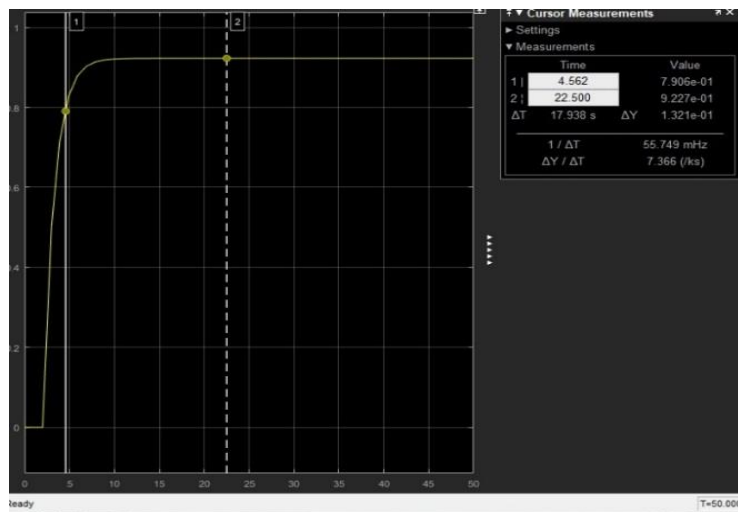


Ilustración 56. Respuesta modelo analítico caso 1 del motor DC 2



Se observa que el comportamiento del modelado no corresponde al del motor y los valores obtenidos distan en gran medida del comportamiento real del motor.

Debido a esto se descarta este modelado y se sigue con los demás modelos.

Simplificación de las ecuaciones del motor DC por el método de eliminación de polo no dominante.

En este caso se usó los datos anteriormente hallados para calcular los polos y la constante que dará una función de transferencia de primer orden de la siguiente forma:

Ilustración 57. Respuesta del modelo del motor DC 1 simplificado

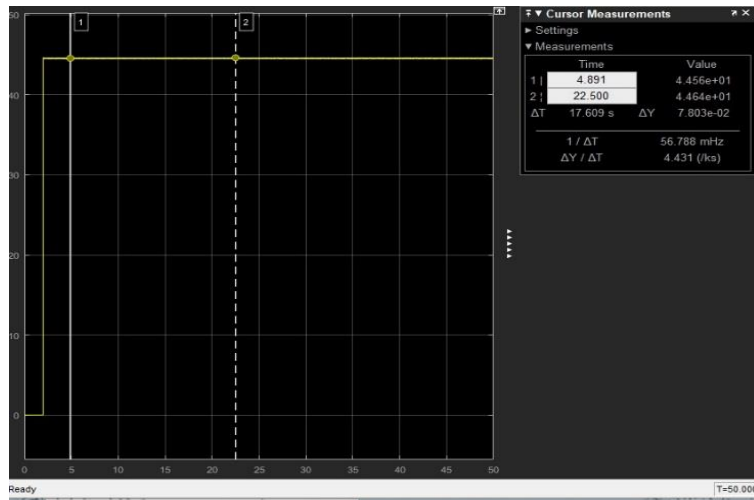


Ilustración 58. Respuesta del modelo del motor DC 2 simplificado



Se observa que el comportamiento de este modelo simplificado no corresponde al comportamiento del motor DC y por lo tanto se descarta su uso para sintonizar el controlador del motor.

Modelo experimental del Motor DC Caso 1

En este caso se usan los datos requeridos para poder representar un modelo experimental tomando velocidades angulares, voltajes de entrada y el cálculo de polos siguiendo las indicaciones del modelado. El comportamiento de este modelo es el siguiente:

Ilustración 59. Respuesta modelo experimental caso 1 del motor DC 1

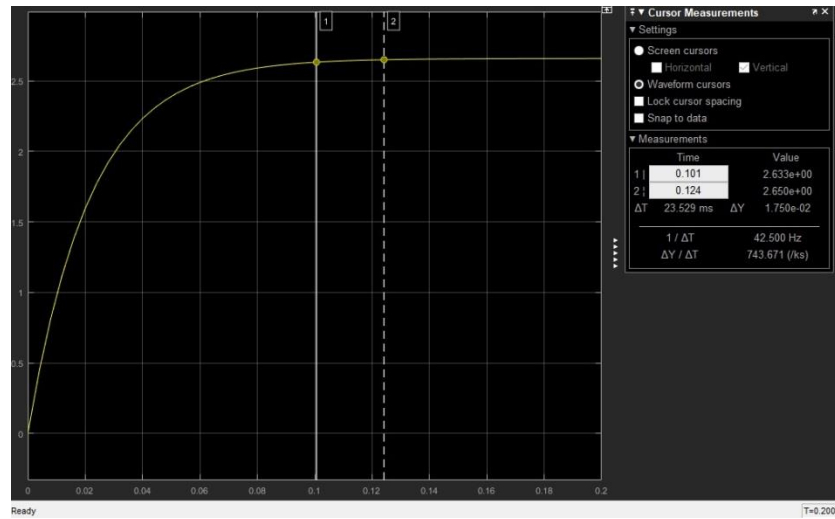
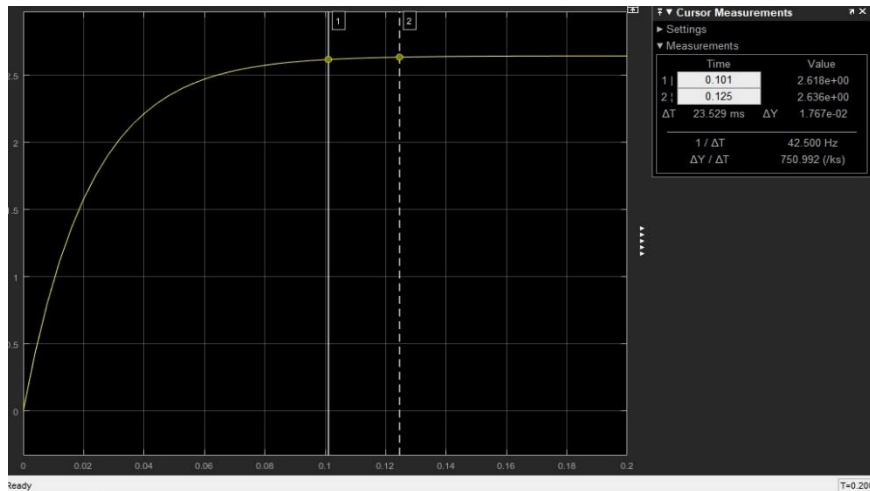


Ilustración 60. Respuesta modelo experimental caso 1 del motor DC 2



Se observa que el comportamiento del modelo se asemeja de forma satisfactoria a la respuesta del motor real ante la entrada de escalón de un voltaje y la salida de velocidad angular, debido a la respuesta satisfactoria en un rango de voltajes de entrada amplia.

Debido a este comportamiento se considera usar este modelado para sintonizar un controlador PID.

Modelo de motor DC usando la herramienta Ident de Matlab

La toma de datos para identificar el modelo del motor DC fue con más muestras debido a la necesidad de datos para hacer comparaciones entre entradas y salidas hechas por la herramienta para dar la función de transferencia. Esta herramienta permite colocar el número de zeros y polos que se quieren para dar la función de transferencia, de la siguiente forma:

Ilustración 61. Modelado de los motores DC Ident

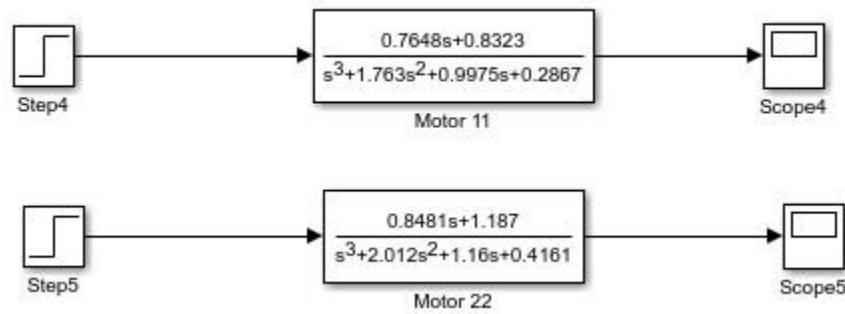


Ilustración 62. Respuesta modelo ident Motor DC 1

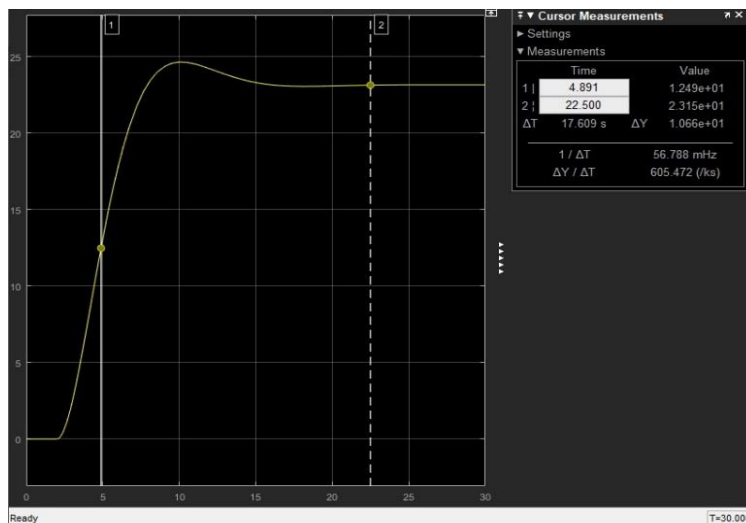
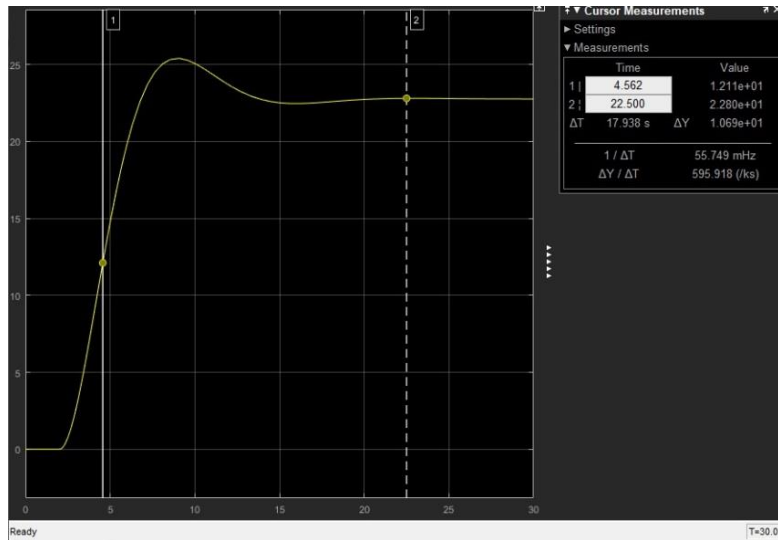


Ilustración 63. Respuesta modelo ident Motor DC 2



Se observa que el comportamiento también cumple en cierta medida con el comportamiento del motor se concluye que funciona en un rango limitado y su respuesta es demorada.

Análisis de Respuesta de Modelo de los Motores DC

Después de establecer el modelo se analiza el comportamiento del sistema, esto se puede realizar mediante diferentes métodos, para poder establecer sistemas de control hay que tener una base de funcionamiento del sistema por medio de señales de entrada y comparando las respuestas.

Algunos criterios de diseño de reguladores se basan en la respuesta del sistema a los cambios de las condiciones iniciales.

Señal de Prueba

Las señales de entrada de prueba más comunes son las de escalón, funciones rampa, funciones sinusoidal, entre otras, con estas se pueden realizar análisis experimentales y matemáticos a los sistemas de manera muy sencilla, ya que, estas señales son funciones de tiempo simples.

La señal que fue utilizada para este análisis fue la de función escalón, ya que, el sistema va a tener perturbaciones que requerirán de una respuesta brusca por parte de los motores dc para

cumplir con el objetivo de llevar a cabo satisfactoriamente la trayectoria que debe realizar el paciente.

Al observar el comportamiento del sistema ante la entrada de función escalón se deben analizar la respuesta transitoria y la respuesta estacionaria

En este caso el sistema es de primer orden lo cual permitió realizar un análisis más simple que en los sistemas de orden superior permitiendo obtener una gráfica de la respuesta en donde se analiza los tiempos que tarda en llegar el sistema al 63.2%, 86.5%, 95%, 98.2% y 99.3%

Para los sistemas que modelan los motores dc ante la señal de entrada de función escalón las respuestas son las siguientes:

Tabla 10. Respuesta a señal de entrada función escalón de los modelos de Motores DC

% (T)	Motor DC1		Motor DC2	
	Salida	Tiempo(S)	Salida	Tiempo(S)
63.2% (1T)	1.681	0.022	1.667	0.022
86.5% (2T)	2.297	0.044	2.284	0.044
95% (3T)	2.531	0.066	2.514	0.066
98.2% (4T)	2.612	0.088	2.596	0.088
99.3% (5T)	2.642	0.110	2.627	0.110

Se puede determinar que el valor de T es 0.022s.

Ilustración 64. Respuesta a escalón Motor DC 1

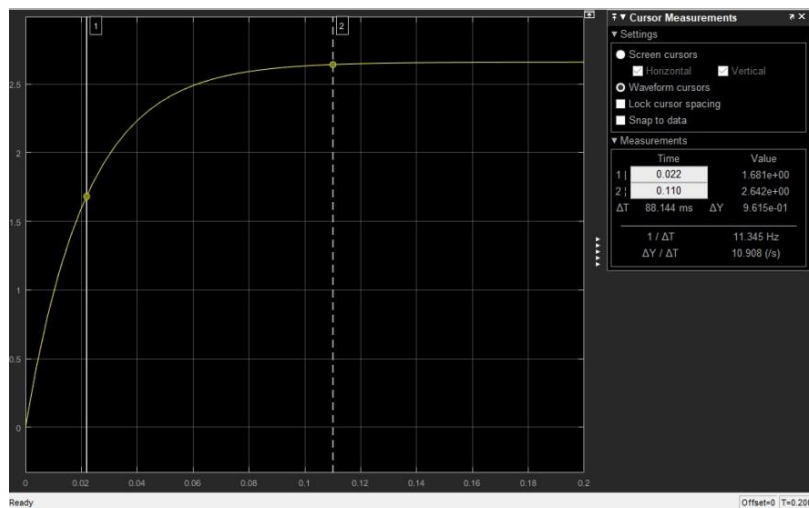
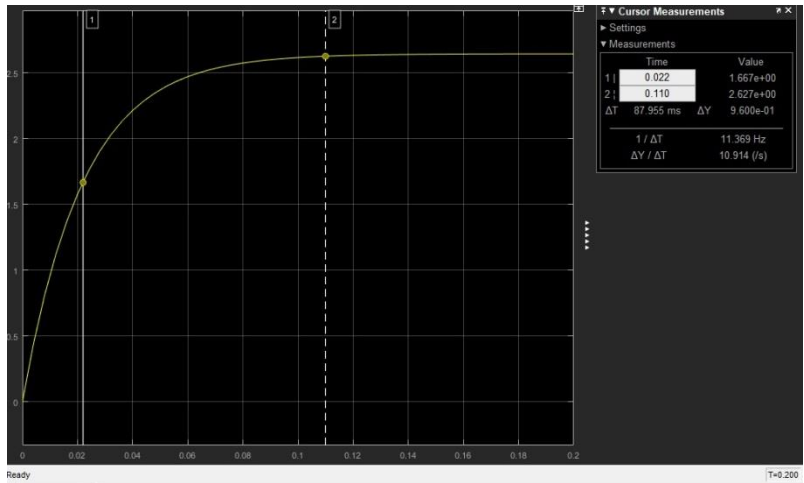


Ilustración 65. Respuesta a escalón Motor DC 2



Análisis del Error

Otras características que se tuvieron en cuenta fueron los errores que se pueden presentar a la hora de observar el comportamiento debido a factores como modificaciones en la entrada de referencia que producen errores en el transitorio y también pueden presentarse en el estacionario esto debido a diferentes factores que pueden ser imperfecciones en los componentes como fricción estática, juego, envejecimiento, entre otros que se pueden ir presentando y que cambian el modelo del sistema.

Sintonización del Regulador PID para controlar los Motores DC

Después de hallar el modelo matemático que representa el comportamiento del motor dc se deben aplicar las técnicas adecuadas para sintonizar un regulador PID que permita controlar cada uno de los motores teniendo en cuenta la respuesta del sistema. Debido a esto se plantea que es posible sintonizar un controlador PID para este sistema, esto por el primer método de Ziegler – Nichols. Este método es gráfico y experimental ya que no ofrece componentes analíticas y se basa en hallar un punto de inflexión de la gráfica que tiene un comportamiento de una S. Teniendo este punto de inflexión se traza una línea tangente a la gráfica para obtener donde corta en el eje de tiempo y así determinar qué valores de L y T se deben tener en cuenta para hallar las

ganancias del proporcional, del integral y el del derivativo, usando la tabla descrita para este método.

Tabla 11. Regla de sintonía de Ziegler-Nichols basada en la respuesta escalón de la planta (primer método).

Tipo de Controlador	K_p	T_i	T_d
P	$\frac{T}{L}$	∞	0
PI	$0.9 \frac{T}{L}$	$\frac{L}{0.3}$	0
PID	$1.2 \frac{T}{L}$	$2L$	0.5L

Después de obtener la gráfica del comportamiento del sistema se hace el análisis del primer método de Ziegler – Nichols y se hallan los valores de T y L.

Ilustración 66. Análisis gráfico del sistema por el primer método de Ziegler-Nichols Motor DC 1.

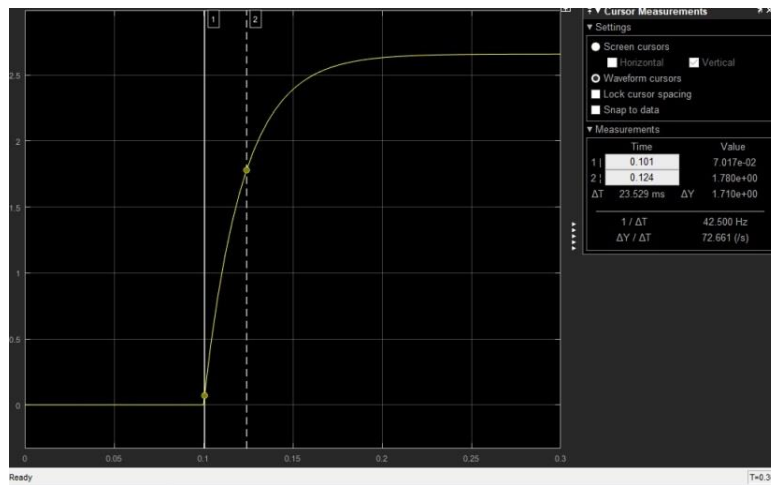
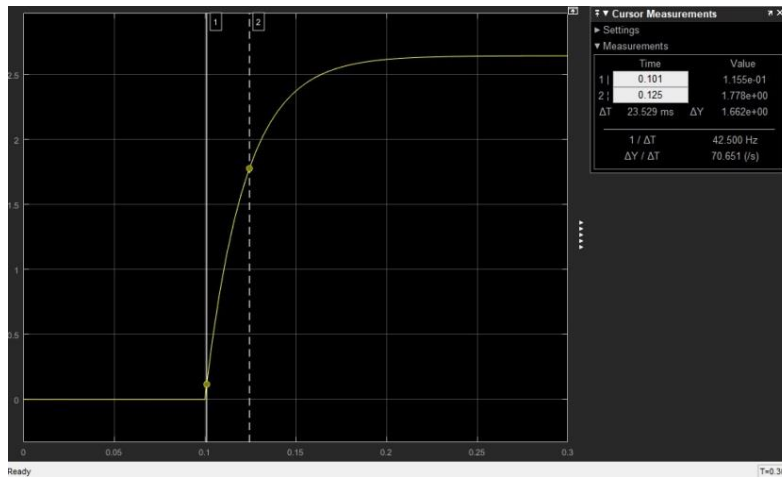


Ilustración 67. Análisis gráfico del sistema por el primer método de Ziegler-Nichols Motor DC 2.



De este análisis se encontraron los valores requeridos los cuales fueron.

Tabla 12. Valores de T y L de los Motores DC

	<i>T</i>	<i>L</i>
Motor DC 1	0.023	0.001
Motor DC 2	0.024	0.001

Con estos valores se halla la función de transferencia del controlador PID con la siguiente fórmula.

$$G(S) = \frac{0.6T \left(S + \frac{1}{L} \right)^2}{S}$$

Control Cinemático

Para poder cumplir con la trayectoria de forma correcta se debe asegurar que los motores dc lleguen a los ángulos requeridos a la misma velocidad debido a que si no lo hacen la trayectoria no se llevará a cabo de forma correcta, para esto se trabajó con las trayectorias coordenadas o isócronas haciendo de este modo que las velocidades de los actuadores tengan que ralentizarse para que al actuar al tiempo estos lleguen al tiempo de los actuadores que tengan menor recorrido por hacer.

3.3. Prototipo robotizado que permita efectuar los movimientos de flexo-extensión de acuerdo a los ejercicios planteados para la terapia de rehabilitación.

Teniendo en cuenta que en los tres objetivos anteriores se llevó a cabo primero el cálculo de las trayectorias, segundo se aplicó la metodología descrita en el objetivo para hallar el diseño del mecanismo que permitió el desarrollo de las trayectorias deseadas dependiendo del paciente y sus medidas antropométricas, posteriormente en el tercero se modeló matemáticamente los motores DC que se controlan para llevar a cabo los movimientos requeridos, en este último objetivo se desarrolla la construcción del prototipo robótico realizando la parte mecánica de control y el ensamblaje electrónico necesario para automatizar el prototipo incluyendo la interfaz hombre - máquina (HMI) que se realizó por medio del internet de las cosas (IoT que precisa para ingresar los datos del paciente y que el robot pueda cumplir con la trayectoria.

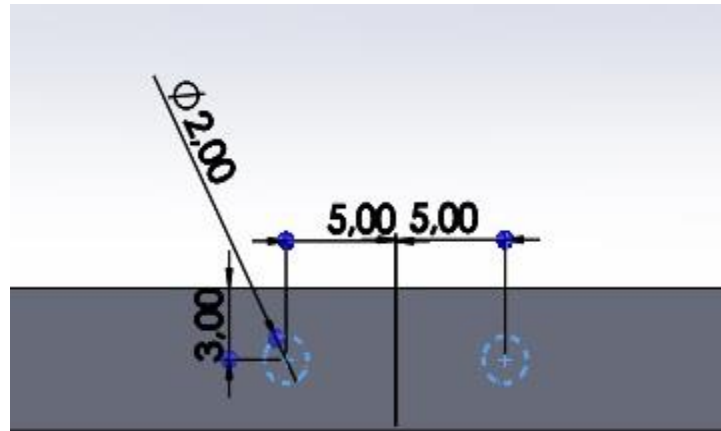
En este caso es necesario hacer construcción mecánica y electrónica que permita al prototipo robótico hacer los ejercicios de flexo-extensión que son recomendados por los fisioterapeutas. Para esto se requirieron de diferentes materiales y recursos, usando varios software para apoyar el diseño de cada una de las piezas, programación y determinación de las trayectorias deseadas.

Construcción de la estructura mecánica del prototipo

Por medio de un software CAD se realiza el diseño de la estructura mecánica y del mecanismo hallado con las dimensiones calculadas que permitan llevar a cabo los movimientos necesarios para los ejercicios de rehabilitación, que en este caso son de flexo-extensión de la articulación de la rodilla en posición supino y con tobillo bloqueado, teniendo en cuenta que las camillas usadas por los fisioterapeutas tienen unas dimensiones estándar mínimas las cuales deben tenerse en cuenta a la hora de dimensionar la base del mecanismo, estas medidas de la camilla son: 180 cm de largo, 60 cm de ancho y 70 cm de alto; medidas que son importantes para determinar la altura a la que debe estar el mecanismo para que el paciente en posición supino pueda hacer flexión o extensión de la rodilla como lo haría un profesional de la fisioterapia.

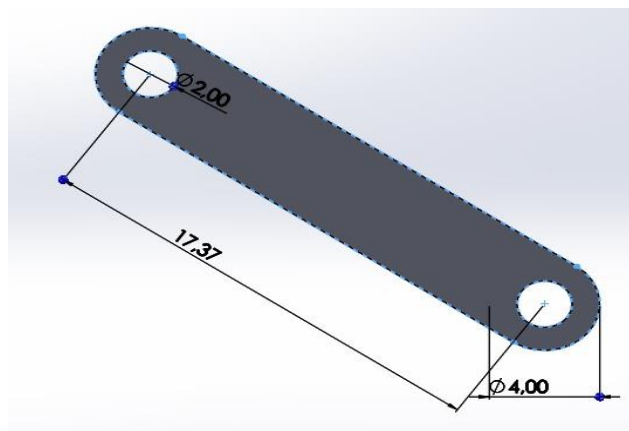
Debido a las consideraciones anteriores y sabiendo que algunas camillas pueden variar sus dimensiones se decidió que la base del mecanismo tuviese una altura ajustable para poder acomodar el prototipo de la forma mas conveniente posible par que el paciente tenga comodidad y el profesional pueda realizar su trabajo de forma cómoda.

Ilustración 68. Medida de Separación entre los ejes de los motores



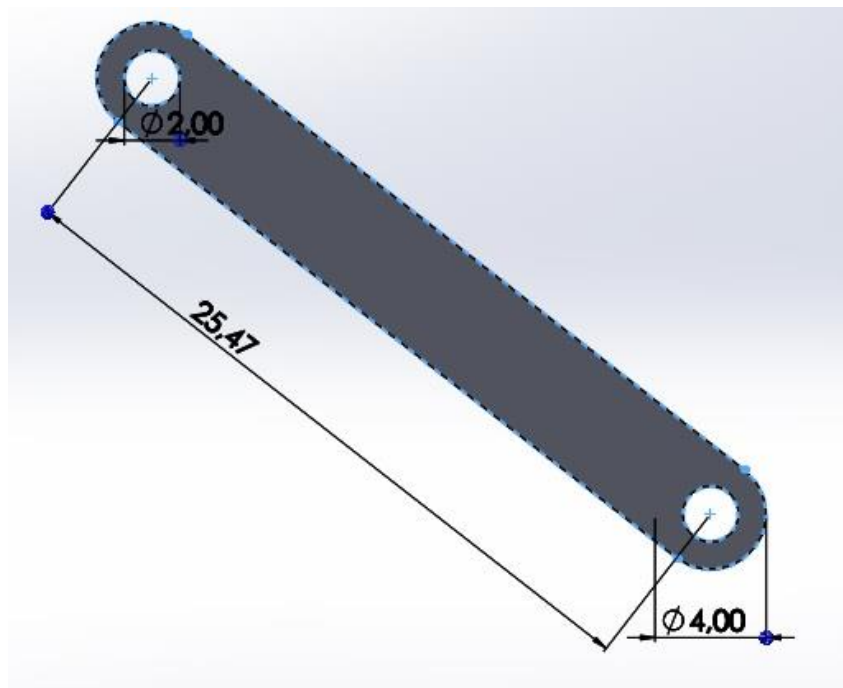
Después de realizar la optimización del mecanismo en donde se encontraron las medidas más apropiadas teniendo en cuenta las dimensiones del mecanismo se encontró que la separación entre los ejes de los motores que cumplían el objetivo de trazar la familia de trayectorias era de 10cm, en este caso se tuvo en cuenta el tamaño de los motores para que estos no colisionaran entre si.

Ilustración 69. Eslabón binario que conecta a los ejes de los motores



Este eslabón binario que conecta directamente una de sus juntas con el eje de salida del motor y la otra junta con el eslabón binario que conecta al ternario, cuenta con dos agujeros de 2 cm de diámetro en donde se localizaran rodamientos que permitan la rotación, una distancia entre ejes de las juntas de 17,37 cm esta distancia fue determinada por la optimización del mecanismo y para la construcción de este se determinó que eran necesarios 4 de estos eslabones con el fin de que la estructura contara con un diseño el cual soportara los esfuerzos de su tarea sin que presentara flexión y por consiguiente esfuerzos perpendiculares en el eje de salida del motor.

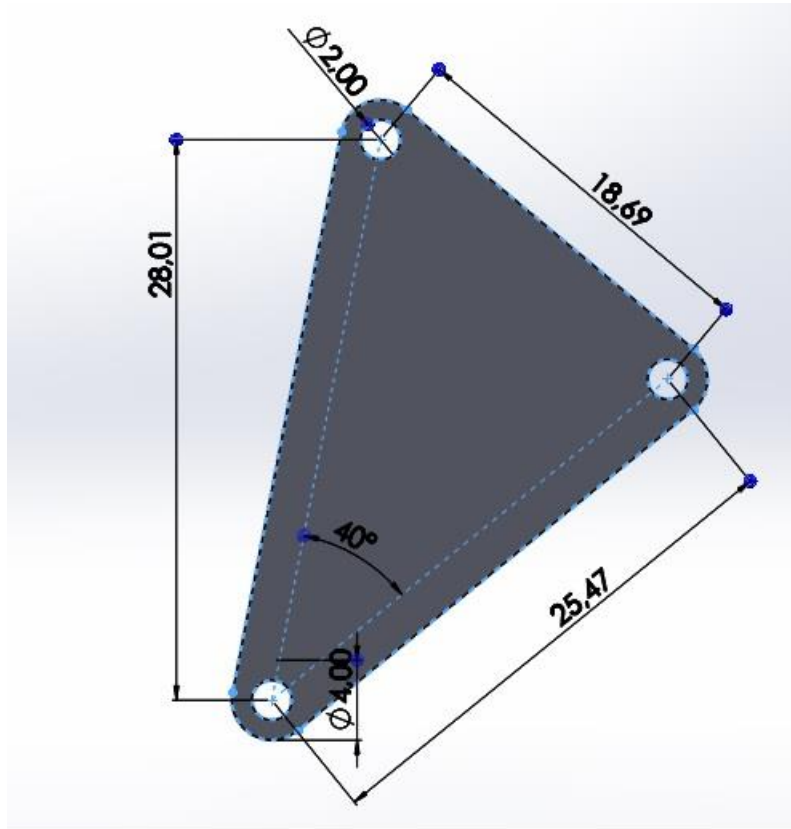
Ilustración 70. Eslabón binario que conecta al eslabón ternario



Este eslabón binario que conecta directamente una de sus juntas con el eje binario que va al motor y la otra junta que va al eslabón ternario, cuenta con dos agujeros de 2 cm de diámetro en donde se localizaran rodamientos que permitan la rotación, tiene una distancia entre ejes de las juntas de 25,47cm esta distancia fue determinada por la optimización del mecanismo y para la construcción de este se determinó que eran necesarios 2 de estos eslabones con el fin

de que la estructura contara con un diseño el cual soportara los esfuerzos de su tarea sin que presentara flexión y por consiguiente esfuerzos perpendiculares en el eje de salida del motor.

Ilustración 71. Eslabón ternario que conecta a la pierna del paciente



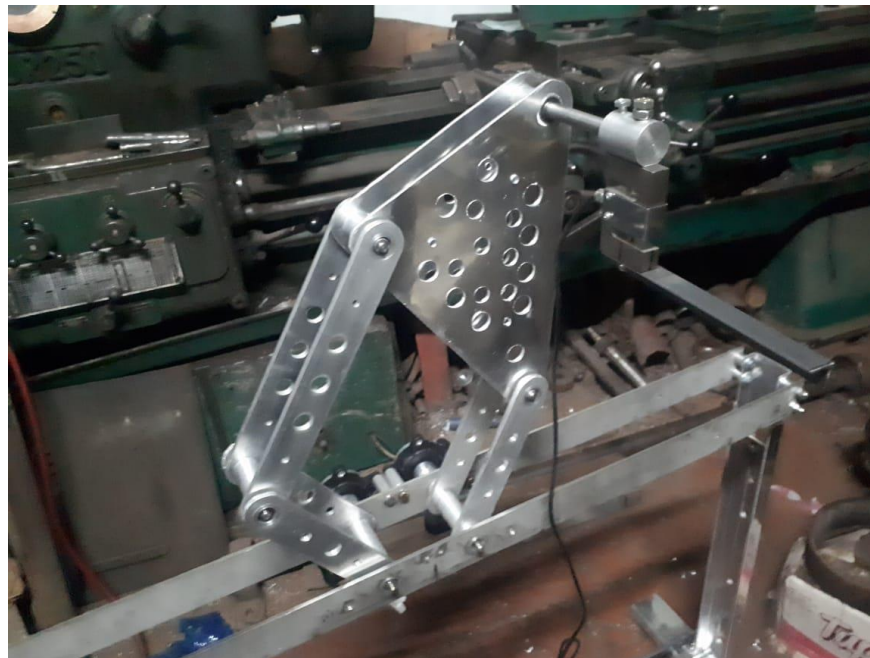
Este eslabón ternario que conecta en dos de sus juntas a los eslabones del mecanismo mencionados anteriormente, y la otra junta que va a un eje el cual va a tener el sensor de carga que mide la fuerza realizada por el paciente y este va a poseer un cala pie el cual va a ajustar el pie del paciente, este será el punto de conexión del mecanismo con la pierna tratando de tener el tobillo a la medida en la que se encuentre el eje de salida del eslabón ternario. Cuenta con tres agujeros de 2 cm de diámetro en donde se localizaran rodamientos que permitan la rotación, tiene una distancia entre ejes de las juntas que van conectados a los otros dos eslabones binarios de 25,47cm que es igual a la distancia entre ejes de uno de los eslabones binarios. La otra junta que conecta con la pierna está ubicada a 40 grados de una línea recta entre los ejes de la distancia anteriormente descrita y a una distancia de 28,01 cm, estas distancias fueron

determinadas por la optimización del mecanismo y para la construcción de este se determinó que eran necesarios 2 de estos eslabones con el fin de que la estructura contara con un diseño el cual soportara los esfuerzos de su tarea sin que presentara flexión y por consiguiente esfuerzos perpendiculares en el eje de salida del motor.

Se aclara que el material utilizado para construir el mecanismo es aluminio, debido a la relación peso-resistencia que este material ofrece, otro material usado fue el acero inoxidable debido a sus características que ayudan a que la estructura sea fuerte. Estos materiales son utilizados en toda la base del mecanismo.

En la siguiente Figura se observan los materiales utilizados:

Ilustración 72. Materiales utilizados



Construcción de la electrónica de control del prototipo

Para la construcción de este circuito electrónico fueron necesarios tarjeta de desarrollo wifi, tarjeta de desarrollo Arduino, drivers para los motores, conversor de nivel lógico, conversor analógico digital, sensor de carga, encoder incremental de alta resolución, fuente de entrada de 110 V AC y salida de 12 V DC de tres canales a 20 A, Fuentes regulables DC a DC y

Motores DC, que permitieron generar la interfaz hombre maquina (HMI) usando un navegador web para el envío de los datos requeridos por medio de comunicación SPI hacia la tarjeta de desarrollo Arduino la cual procesará estos datos realizando los cálculos necesarios para determinar los arcos de movilidad de la rodilla en el paciente y los ángulos en que deben posicionarse los motores, después de este proceso esta misma tarjeta se encarga de enviar datos hacia los drivers de los motores DC que los controlaran haciendo de esta forma los movimientos necesarios.

Ilustración 73. Tarjeta ESP32 WIFI



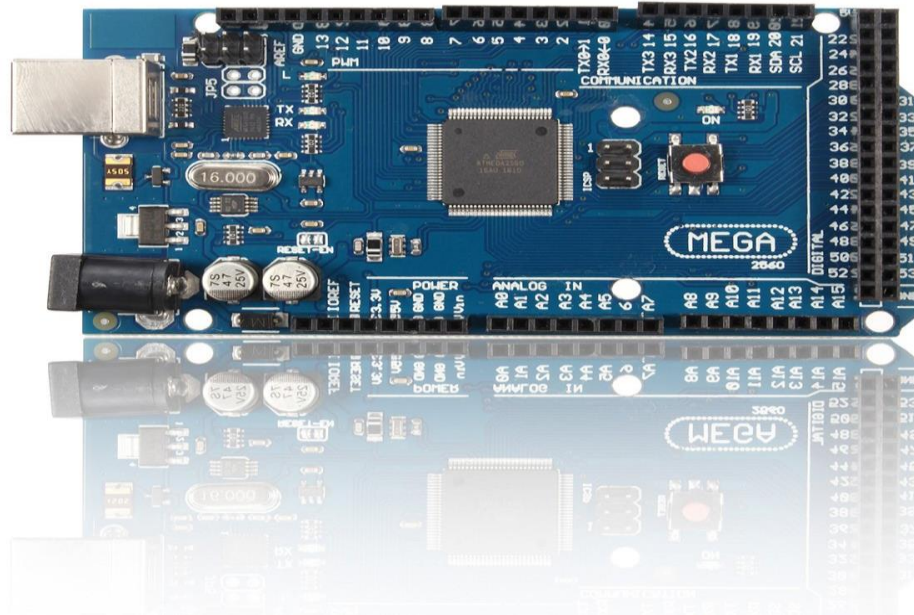
Esta tarjeta de desarrollo ESP32 cuenta con el módulo wifi que permite la conexión con una red de internet la cual tienen varias formas de uso, en este caso se utilizó esta tarjeta como un web server que contiene el diseño de una página web HTML, esta tarjeta se conecta a una red wifi por medio del ID y contraseña de la red, al conectarse esta genera una IP propia la cual puede accederse en cualquier navegador de cualquier dispositivo que tenga conexión a esta misma red de internet.

Esto se hizo con el fin de que el usuario del prototipo pueda enviar los datos hacia el robot sin necesidad de contar con una aplicación o software específico, sino usando las herramientas que cualquier dispositivo móvil o PC cuenta, para evitar inconvenientes de costos o recursos requeridos por los dispositivos. Esta tarjeta también se comunica con la tarjeta de desarrollo que realizara los cálculos y control de motores pertinentes por medio del protocolo SPI (Serial

Peripheral Interface) para enviar los datos ingresados por el usuario por la página web para que este haga el procesamiento requerido.

Interfaz hombre máquina (MHI) del prototipo

Ilustración 74. Tarjeta de desarrollo Arduino Mega 2560



En esta tarjeta de desarrollo Arduino Mega 2560, se llevara a cabo todo el procesamiento de datos ingresados por el usuario por medio de la página web para obtener los ángulos a los cuales los motores del prototipo deben posicionarse para cada trayectoria, también a esta deben ir conectado los sensores para realizar la retroalimentación que servirán para medir posición de motores y esfuerzo hecho por el paciente, también tendrá el objetivo de controlar los drivers de los motores DC para que estos funcionen correctamente.

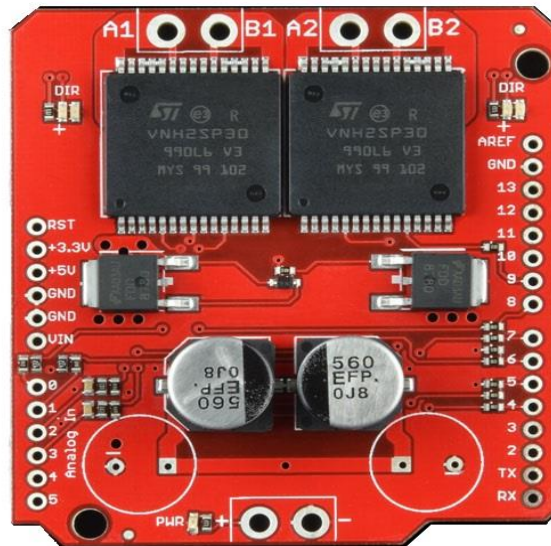
Para la recepción de los datos enviados por el usuario se tiene una comunicación por protocolo SPI (Serial Peripheral Interface) con la tarjeta ESP32, estos datos recibidos serán:

- La medida de cadera a rodilla
- La medida de rodilla a tobillo
- Peso del paciente
- Tipo de ejercicio de rehabilitación (pasivo o activo)

- Número de repeticiones.
- Ángulo de arco de movilidad del paciente.

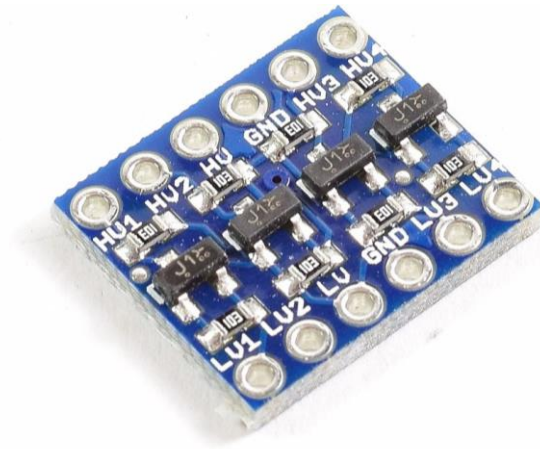
Después de recibir los datos realiza los cálculos necesarios para hallar los ángulos en los que los motores deben posicionarse para cumplir con la trayectoria, seguido a esto se realiza el control de los motores DC por medio de un controlado PID ajustado en la tarjeta enviando por medio de pulso PWM y pines digitales la velocidad y activación de los motores a los drivers que lo controlan al tiempo que captura la medida de los sensores para retroalimentar su funcionamiento.

Ilustracion 75. Drivers para motores DC Monster Moto Shield Chip VNH2SP30



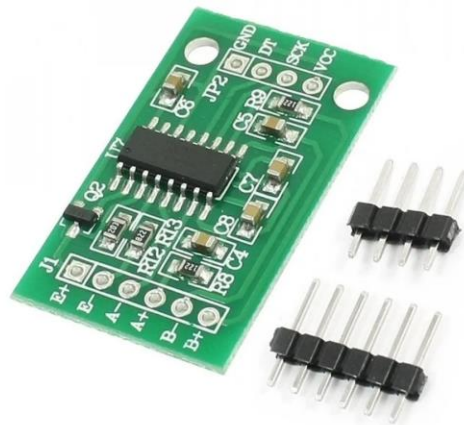
Esta tarjeta contiene dos puente H que sirven para controlar velocidad y encendido de motores DC con unas tolerancias de voltaje de entrada de motores hasta 24 V y tolerancias de corriente nominal de 14 A y corriente pico de 30 A, sus niveles lógicos de voltaje son los mismo que en una tarjeta Arduino.

Ilustración 76. Conversor de nivel lógico



Este circuito sirve como puente en la comunicación por protocolo SPI entre la tarjeta ESP32 y la Arduino mega 2560 debido a que la primera trabaja a niveles de voltaje de 3.3. V y la segunda a 5 V, y para evitar daños en la tarjeta que tiene el módulo wifi.

Ilustración 77. Conversor analógico digital HX711



Este conversor se usa para conectar el sensor de carga para transformar los datos que este envía y que puedan ser leídos por la arduinno mega 2560.

Ilustración 78. Sensor de carga Tipo S.



El sensor de carga tipo S que se muestra en la ilustración 78 fue escogido debido a su construcción y funcionamiento, ya que, este mide la fuerza ejercida sobre el mismo tanto en la compresión como en la extensión. Este tipo de sensores no pueden ser conectados directamente a la tarjeta de adquisición de datos, en este caso el sensor requiere de un driver como el que se muestra en la ilustración 77 el cuál se encarga de convertir las señales provenientes del sensor y también se encarga de la alimentación del mismo.

Ilustración 79. Encoder incremental de alta resolución



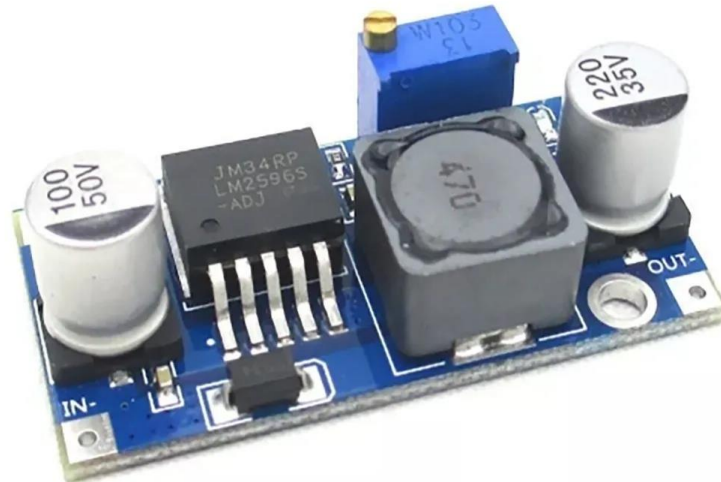
Estos sensores mostrados en ilustración 79 tienen dos canales para lecturas de pulso los cuales determinan el ángulo al que los motores se van a encontrar, este encoder incremental tiene una resolución de 600 pulsos por vuelta ofreciendo 0,6 grados de precisión de medición de ángulo, este va conectado directamente a los pines de interrupciones del Arduino mega 2560 pero la alimentación de este es externa debido a que su nivel de voltaje es de 6 a 12 V y por lo tanto no puede ser alimentado con los pines del Arduino.

Ilustración 80. Fuente de entrada de 110 V AC y salida de 12 V DC de tres canales a 20 A



Esta fuente que se muestra en la ilustración 80 se utiliza para alimentar todo el circuito electrónico y alimentar los motores del prototipo. En este caso se pensó en esta fuente debido al consumo previsto por los motores principalmente, en especial, su voltaje de alimentación de 12V y un consumo de corriente pico de 7A en cada motor DC y esto sumado con el consumo del circuito electrónico, en voltaje la alimentación de estos dispositivos fueron de diferentes valores inferiores a los 12V, así que, se usaron reguladores de voltaje para alimentarlos y analizando el consumo de estos en corriente el cuál combinado no es muy alto y la fuente puede perfectamente alimentar el robot.

Ilustración 81. Fuentes regulables DC a DC



Para el circuito se necesitan tres reguladores de este tipo, para alimentar la Arduino mega 2560, otra para alimentar el incoder incremental y la última para alimentar la tarjeta ESP 32, esto debido a que cada uno trabaja con voltajes y corrientes diferentes.

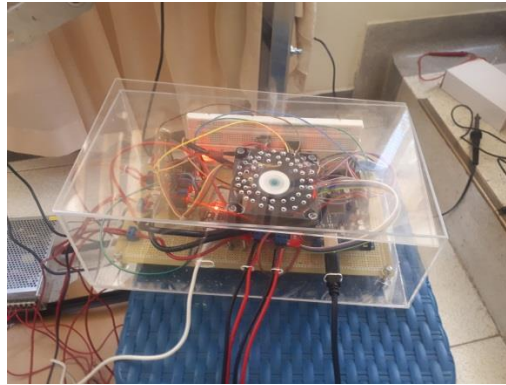
Ilustración 82. Motores DC



Los motores mostrados en la ilustración 82 son necesarios para poder mover el mecanismo, estos fueron elegidos debido a dos situaciones, primero que después de hechos los cálculos de los pares necesarios ante unos pares mínimos a la entrada del mecanismo se consideró que pueden cumplir con estos pares y lo segundo es debido a que su construcción de tornillo sin fin corona proporciona un alto coeficiente de fricción que hace que al quedar des energizados

estos quedan en la misma posición por lo tanto se puede decir que se desprecian las perturbaciones obtenidas por la carga en el cala pie del mecanismo.

Ilustración 83. Placa electrónica.



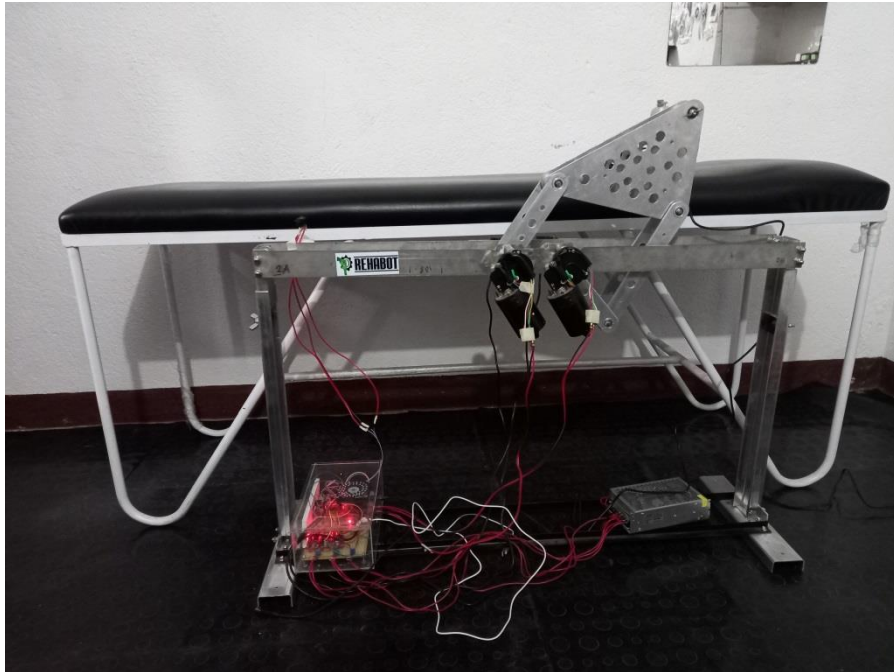
A partir de la utilización de los elementos anteriormente mencionados se conformó la placa electrónica para realizar la interfaz, recepción de datos, envío de datos, control de motores y Realimentación por medio de los sensores para el funcionamiento del prototipo robótico.

Ilustración 84. Logo y nombre del prototipo robótico



A partir del resultado del trabajo práctico se diseñó un logo para el prototipo robótico que simboliza la tecnología e ingeniería combinada con la salud y específicamente con la rehabilitación.

Ilustración 85. Prototipo terminado



Este fue el resultado final del prototipo integrando todos los componentes mecánicos y electrónicos.

CAPÍTULO IV

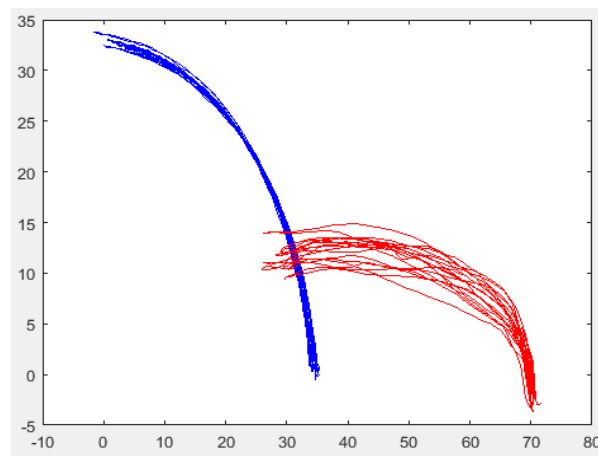
Resultados

Se realizó un análisis biomecánico del movimiento de la rodilla en posición supino, fueron hechos y supervisados por profesionales de la fisioterapia, quienes tomaron a un paciente como referencia para poder realizar el análisis, este análisis se realizó colocando unas marcas en las articulaciones de los miembros inferiores (cadera, rodilla y tobillo) ejecutando repeticiones de estos movimientos siendo capturados en video para posteriormente determinar la trayectoria requerida, en este caso se determinó que el punto de referencia era el tobillo puesto que el fisioterapeuta toma al paciente de esta área.

Ilustración 86. Análisis biomecánico del paciente referencia.



Ilustración 87. Análisis de los datos tomados.



A partir de hallar la trayectoria y el punto de referencia en la rodilla surgía la limitante de tener que realizar las mediciones a cada persona, es decir el mecanismo debía estar en la rodilla y generaba problema al cambiar del paciente por las dimensiones de la pierna, al revisar los videos de ejercicios de rehabilitación se observa que se utiliza el tobillo bloqueado, por lo que se descubre que por biomecánica al tomar el tobillo como punto fijo podría la rodilla seguir la trayectoria deseada, así mismo, no se necesitaba analizar a todas las persona, sino que al tener un paciente referente se sacaba una ecuación con la trayectoria y con los cálculos de las medidas de las piernas de los demás pacientes se encontraría la trayectoria personalizada dependiendo de estas dimensiones.

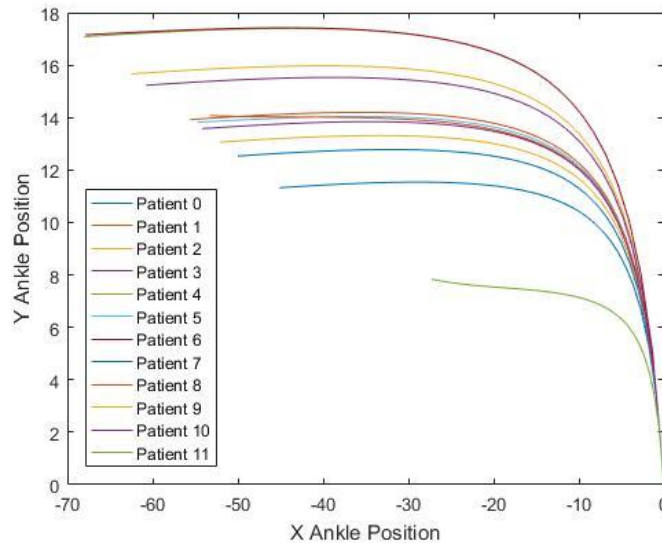
Tabla 13. Puntos en el eje x y eje y con el ángulo de arco de movilidad del paciente referencia.

PACIENTE REFERENCIA		
Arco de Movilidad (Deg)	Px(cm)	Py(cm)
0	0	0.2074
18.1643	-1	3.7191
25.3340	-2	5.8928
30.9373	-3	7.3645
35.7750	-4	8.4225
40.1255	-5	9.2159
44.1238	-6	9.8302
47.8493	-7	10.3174
51.3538	-8	10.7112
54.6742	-9	11.0344
57.8381	-10	11.3029
60.8666	-11	11.5282
63.7767	-12	11.7188
66.5820	-13	11.8809
69.2937	-14	12.0196
71.9215	-15	12.1388
74.4732	-16	12.2413
76.9557	-17	12.3298
79.3750	-18	12.4061
81.7362	-19	12.4720
84.0438	-20	12.5286
86.3017	-21	12.5773
88.5136	-22	12.6189
90.6824	-23	12.6543

92.8110	-24	12.6841
94.9018	-25	12.7089
96.9573	-26	12.7292
98.9792	-27	12.7456
100.9696	-28	12.7582
102.9300	-29	12.7676
104.8620	-30	12.7740
106.7668	-31	12.7776
108.6457	-32	12.7787
110.4999	-33	12.7776
112.3303	-34	12.7743
114.1378	-35	12.7690
115.9231	-36	12.7620
117.6871	-37	12.7533
119.4303	-38	12.7430
121.1532	-39	12.7313
122.8562	-40	12.7183
124.5398	-41	12.7041
126.2041	-42	12.6886
127.8493	-43	12.6721
129.4756	-44	12.6546
131.0828	-45	12.6361
132.6708	-46	12.6167
134.2394	-47	12.5965
135.7881	-48	12.5755
137.3165	-49	12.5537
138.8238	-50	12.5313

Después de encontrar la trayectoria en el paciente de referencia, se realizan los cálculos necesarios para hallar las trayectorias con pacientes de distintas medidas antropométricas de miembros inferiores.

Ilustración 88. Gráfica de las trayectorias del paciente referencia (Paciente 0) y de los pacientes ejemplo.



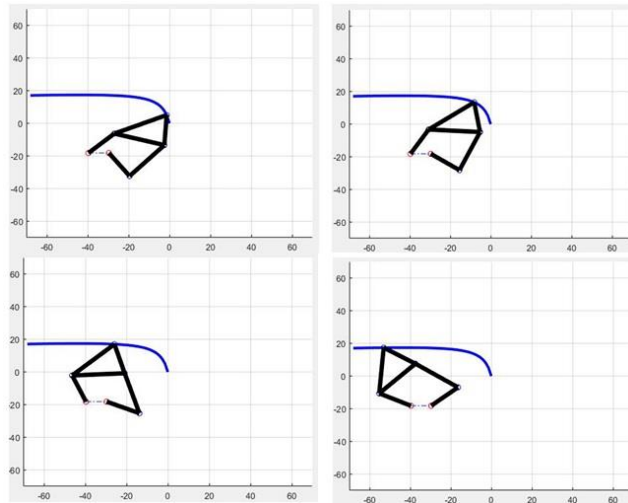
Para encontrar el mecanismo se maneja la metodología de síntesis de mecanismo que cuenta con tres fases: Síntesis de número, síntesis de tipo y síntesis dimensional, hallando dos resultados: un mecanismo sin tener en cuenta la pierna como parte del mecanismo y el segundo, en donde se cuenta la pierna como parte del mecanismo con lo que se concluyó que era el más efectivo y mecanismo que se utilizó. Posteriormente se utilizaron algoritmos de inteligencia artificial para optimizar las dimensiones del mecanismo teniendo como base encontrar las menores dimensiones posibles.

Tabla 14. Dimensiones obtenidas con la optimización.

Partes del mecanismo	Medida del mecanismo(cm)
L: Distancia entre los dos actuadores del robot	10.0004
L1=L3: Eslabones del mecanismo que se conectan a los actuadores	17.3687
L2=L4: Eslabones del mecanismo	25.4656
L5: Distancia al punto de interés	28.0100

Ángulo hacia el punto de interés	40°
----------------------------------	-----

Ilustración 89. Mecanismo con pierna obtenido con la optimización



Después de la optimización se realiza el análisis cinemático y estático de los dos mecanismos analizados anteriormente para así determinar velocidades angulares y pares necesarios en los motores para que estos puedan llevar a cabo las trayectorias deseadas, para estos cálculos se utilizó el método de Davies que se basa en teoría de grafos, teoría de redes de vectores y helicoides. Para llevar a cabo estos cálculos se tuvo en cuenta el peso de la pierna del paciente, en este caso se tomó al paciente referencia el cuál tiene un peso aproximado de la extremidad de 10 kg. La tabla 15 muestra los pares requeridos por los motores para las dimensiones y peso del paciente referencia. Se aclara que para cada paciente se deben calcular los valores dependiendo de sus medidas y de su peso.

Tabla 15. Pares requeridos en los motores del robot para el paciente referencia.

Ángulo de Arco de Movilidad(Deg)	Par de Motor 1(N.m)	Par de Motor 2(N.m)
0	-25.6290	10.2364
18.1643	-25.7769	10.4650
25.3340	-25.7935	10.5703

30.9373	-25.7170	10.5401
35.7750	-25.5642	10.3920
40.1255	-25.3471	10.1491
44.1238	-25.0736	9.8298
47.8493	-24.7493	9.4486
51.3538	-24.3771	9.0159
54.6742	-23.9617	8.5432
57.8381	-23.5047	8.0383
60.8666	-23.0070	7.5064
63.7767	-22.4725	6.9594
66.5820	-21.9008	6.4012
69.2937	-21.2943	5.8408
71.9215	-20.6570	5.2911
74.4732	-19.9908	4.7616
76.9557	-19.2976	4.2634
79.3750	-18.5819	3.8114
81.7362	-17.8455	3.4190
84.0438	-17.0941	3.1058
86.3017	-16.3306	2.8894
88.5136	-15.5542	2.7836
90.6824	-14.7666	2.8071
92.8110	-13.9667	2.9728
94.9018	-13.1490	3.2886
96.9573	-12.3102	3.7574
98.9792	-11.4451	4.3727
100.9696	-10.5455	5.1143
102.9300	-9.6145	5.9613
104.8620	-8.6604	6.8819
106.7668	-7.6951	7.8419
108.6457	-6.7530	8.8189

110.4999	-5.8568	9.7829
112.3303	-5.0396	10.7189
114.1378	-4.3370	11.6274
115.9231	-3.7715	12.5071
117.6871	-3.3554	13.3600
119.4303	-3.0994	14.1951
121.1532	-2.9965	15.0172
122.8562	-3.0438	15.8333
124.5398	-3.2169	16.6359
126.2041	-3.5108	17.4338
127.8493	-3.8984	18.2175
129.4756	-4.3683	18.9878
131.0828	-4.9032	19.7405
132.6708	-5.4840	20.4689
134.2394	-6.1012	21.1725
135.7881	-6.7420	21.8477
137.3165	-7.3967	22.4911
138.8238	-8.0576	23.1022

Debido al uso de motores de corriente directa para mover el prototipo robótico se hace necesario proponer un control en el cual se relacione una entrada de voltaje regulada por PWM y que tenga como salida una posición angular deseada, para esto es necesario primero obtener el modelo matemático de los motores.

Posterior a obtener el modelo matemático se calcula un controlado clásico PID que permita al motor llegar a los ángulos calculados para cada trayectoria. Después de sintonizado el controlador se ejecuta un control cinemático el cual va a permitirle al robot encontrar una función que le permita llegar de un ángulo a otro ángulo.

Al hacer el análisis de síntesis de mecanismos sin pierna no generaba los mejores resultados para el mecanismo final, por lo que se decidió utilizar la pierna como parte del mecanismo y realizar cálculos con y sin pierna, en los dos se da un mecanismo con 5 barras pero con la diferencia de que existe un eslabón terciario, así se definió el punto de interés en este eslabón, se hizo la optimización de estos mecanismos se determinó que la optimización se haría por dimensiones hasta llegar a las más pequeñas y en cuanto al método de Davies se probaron con pierna y sin pierna y resultó que teniendo en cuenta la pierna se generan dos circuitos y sin pierna uno solo, lo que es un cambio significativo puesto que de esta forma los cálculos son más grandes y complejos pero más eficientes en la cinemática y la estática.

En el tercer objetivo de acuerdo a los resultados de los torques, se escogieron dos motores de corriente directa que se utilizan en los automóviles. Debido que son sistemas no reversibles, de esta manera, en caso que no sean energizados no lograrán moverse, al realizar el modelamiento matemático se analizaron desde varios puntos como la analítica y la experimental, se encontraron mejores resultados en el modelamiento experimental, porque en la analítica fue difícil porque no cuenta con las características del fabricante, sin embargo, se intentaron hacer experimentos utilizando Arduino, multímetro y circuitos ya que no se cuenta con un laboratorio por lo que los resultados no fueron los esperados y se dio un modelamiento experimental con Arduino y midiendo velocidad angular posición angular y de allí se tomaron las herramientas matemáticas para poder determinar ese modelo, luego de hallarlo se halló el controlador clásico PID para que diera mejor respuesta y se cumplió con el objetivo, y se trabajó en control cinemático para que el robot pueda ir de Angulo a Angulo, cumplir su trayectoria utilizando técnicas del control dinámico para hacer interpolación y poder unir estos puntos de la trayectoria.

Finalmente se realiza la estructura del mecanismo que va a ejecutar la trayectoria deseada, se trabajó con elementos resistentes pero livianos como aluminio, acero inoxidable teniendo en cuenta los diseños en Software CAD y se basaron en los cálculos del objetivo dos que dieron las dimensiones del mecanismo, en la parte electrónica se plasman los cálculos hallados, para poder encontrar la trayectoria de cada uno de los pacientes de acuerdo a sus dimensiones, esta parte electrónica consta de una tarjeta puente H para controlar los motores por medio del

PWM, una tarjeta Arduino que hace las veces de tarjeta principal, Encoders incrementales para medir el Ángulo, una celda de carga tipo S que se usa para medir la carga en el mecanismo y la fuerza que hace la persona en el robot entre otros electrónicos. También se realizó una comunicación con Arduino mega con otra tarjeta de desarrollo en este caso SP32 la cual es la encargada del HMI interfaz hombre – máquina, SP32 como web Server que se conecta a internet y se accede a ella por página web y desde allí se pueden enviar los datos del paciente y la información que se quiere que el robot tenga para realizar las trayectoria y ejercicios personalizados teniendo en cuenta los ángulos obtenidos.

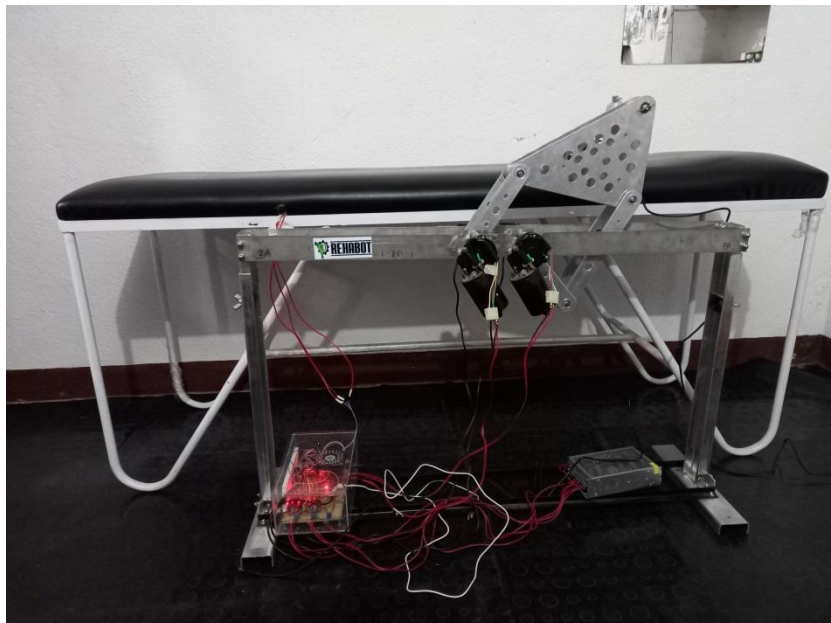
Contando con las dimensiones del robot, los cálculos y el control de motores se realiza la construcción mecánica y electrónica comenzando con un diseño en un software CAD de la estructura del mecanismo, en la parte electrónica se usan diferentes tarjetas electrónicas que permiten la interacción del ser humano y el robot y esta se hace utilizando el internet de las cosas (IoT) que permite por medio de cualquier navegador interactuar con una página web, enviar la información del paciente y de la etapa de rehabilitación para ser procesada y generar los movimientos en el robot.

En la parte eléctrica, para la construcción de este circuito electrónico fueron necesarios tarjeta de desarrollo wifi, tarjeta de desarrollo Arduino, drivers para los motores, conversor de nivel lógico, Conversor analógico digital, sensor de carga, encoder incremental de alta resolución, fuente de entrada de 110 V AC y salida de 12 V DC de tres canales a 20 A, Fuentes regulables DC a DC y Motores DC, que permitieron generar la interfaz hombre maquina (HMI) usando un navegador web para el envío de los datos requeridos por medio de comunicación SPI hacia la tarjeta de desarrollo Arduino la cual procesará estos datos realizando los cálculos necesarios para determinar los arcos de movilidad de la rodilla en el paciente y los ángulos en que deben posicionarse los motores, después de este proceso esta misma tarjeta se encarga de enviar datos hacia los drivers de los motores DC que los controlaran haciendo de esta forma los movimientos necesarios.

En la construcción mecánica, se tuvo que realizar por medio de un software CAD el diseño de la estructura mecánica y del mecanismo hallado con las dimensiones calculadas que permitan

llevar a cabo los movimientos necesarios para los ejercicios de rehabilitación, que en este caso son de flexo-extensión de la articulación de la rodilla en posición supino y con tobillo bloqueado, teniendo en cuenta que las camillas usadas por los fisioterapeutas tienen unas dimensiones estándar mínimas las cuales deben tenerse en cuenta a la hora de dimensionar la base del mecanismo, estas medidas de la camilla son: 180 cm de largo, 60 cm de ancho y 70 cm de alto; medidas que son importantes para determinar la altura a la que debe estar el mecanismo para que el paciente en posición supino pueda hacer flexión o extensión de la rodilla como lo haría un profesional de la fisioterapia.

Ilustración 90. Prototipo del robot construido.



Debido a las consideraciones anteriores y sabiendo que algunas camillas pueden variar sus dimensiones se decidió que la base del mecanismo tuviese una altura ajustable para poder acomodar el prototipo de la forma más conveniente posible para que el paciente tenga comodidad y el profesional pueda realizar su trabajo de forma cómoda.

Al momento de realizar las pruebas con el robot se obtuvieron mediciones con las que se miden los errores se muestra a continuación una captura de un video el cuál se compara con los valores dados.

Ilustración 91. Captura del funcionamiento del robot con el paciente.



En la imagen anterior se puede observar que el ángulo del arco de movilidad al que se quiere llegar en la rodilla es de 60° , usando el software Kinovea se obtiene el ángulo que se encuentra entre la cadera - rodilla y la rodilla - tobillo, en este caso da 122° . El ángulo de arco de movilidad como se vio en el desarrollo del libro es igual a $180^\circ - 122^\circ$. Por lo tanto el ángulo del arco de movilidad probado es de 58° , entonces, el error en esta medición es de 2° .

El prototipo diseñado y construido presenta unas características especiales debido a los materiales y conceptos usados para su construcción mecánica, a la integración de electrónica necesaria para llevar a cabo su funcionamiento, la implementación de IoT (Internet of Things) para la comunicación e interfaz hombre máquina (HMI) y los conocimientos requeridos para realizar el control de sus motores, haciendo de esta forma posible cumplir los objetivos planteados para el presente trabajo.

A la hora de observar el comportamiento del prototipo se observa que este tiene posibilidades de realizar movimientos diferentes para los que fue diseñado y estos podrían servir para la rehabilitación de otras partes del cuerpo bien sea en las articulaciones de los miembros inferiores o en las articulaciones de miembros superiores.

Debido a la construcción del prototipo y a las herramientas usadas se pueden prever trabajos futuros que ayuden a mejorar la interacción de los profesionales de la salud con este dispositivo,

ya que tanto en su estructura mecánica como en su diseño electrónico da la posibilidad de agregar aplicaciones o elementos que pueden dar mayor realce a este trabajo. Un ejemplo o idea para esto es la de agregar una tarjeta de desarrollo WiFi, que tenga integrada una cámara para que se pueda generar un streaming que permita a un profesional de la salud controlar el dispositivo de forma remota y que este pueda observar el desempeño del prototipo con los pacientes, agregar una pantalla donde se puedan observar los diferentes datos que se calculan dentro de la programación, incluyendo IP del dispositivo WiFi para observar la HMI, entre otras consideraciones que se pueden tomar.

Conclusiones

A partir de hallar la trayectoria y el punto de referencia en la rodilla surgía la limitante de tener que realizar las mediciones a cada persona, es decir el mecanismo debía estar en la rodilla y generaba problema al cambiar del paciente por las dimensiones de la pierna, al revisar los videos de ejercicios de rehabilitación se observa que se utiliza el tobillo bloqueado, por lo que se descubre que por biomecánica al tomar el tobillo como punto fijo podría la rodilla seguir la trayectoria deseada, así mismo, no se necesitaba analizar a todas las persona, sino que al tener un paciente referente se sacaba una ecuación con la trayectoria y con los cálculos de las medidas de las piernas de los demás pacientes se encontraría la trayectoria personalizada dependiendo de estas dimensiones.

Al hacer el análisis de síntesis de mecanismos sin pierna no generaba los mejores resultados para el mecanismo final, por lo que se decidió utilizar la pierna como parte del mecanismo y realizar cálculos con y sin pierna, en los dos se Da un mecanismo con 5 barras pero con la diferencia de que existe un eslabón ternario, así se definió el punto de interés en este eslabón, se hizo la optimización de estos mecanismos se determinó que la optimización se haría por dimensiones hasta llegar a las más pequeñas y en cuanto al método de Davies se aplicaron con pierna y sin pierna y resulto que teniendo en cuenta la pierna se generan dos circuitos y sin pierna uno solo, lo que es un cambio significativo puesto que de esta forma los cálculos son más grandes y complejos pero más eficientes en la cinemática y la estática, para este trabajo se utilizaron las medidas y el peso del paciente referencia, de esta forma, como se expresó a lo largo del documento cada vez que un paciente use el prototipo es necesario realizar los cálculos tanto de la trayectoria como de los pares requeridos por los motores.

En el tercer objetivo de acuerdo a los resultados de los torques, se escogieron dos motores eleva vidrios de corriente directa que se utilizan n los automóviles debido a que tienen una construcción en su reducción mecánica de sinfín corona lo que da una ventaja en cuanto a que presentan una fricción muy alta por lo que sin no son energizados no logran moverse, no es la mejor fricción pero en costo beneficio son buenos, al realizar el modelamiento matemático se analizaron desde varios puntos como la analítica y la experimental, se

encontraron mejores resultados en el modelamiento experimental, porque en la analítica fue difícil porque no cuenta con las características del fabricante, sin embargo, se intentaron hacer experimentos utilizando Arduino, multímetro y circuitos ya que no se cuenta con un laboratorio por lo que los resultados no fueron los esperados y se dio un modelamiento experimental con Arduino y midiendo velocidad angular posición angular y de allí se tomaron las herramientas matemáticas para poder determinar ese modelo, luego de hallarlo se halló el controlador clásico PID para que diera mejor respuesta y se cumplió con el objetivo, y se trabajó en control cinemático para que el robot pueda ir de ángulo a ángulo, cumplir su trayectoria utilizando técnicas del control dinámico para hacer interpolación y poder unir estos puntos de la trayectoria.

Se realiza la estructura del mecanismo que va a ejecutar la trayectoria deseada, se trabajó con elementos resistentes pero livianos como aluminio, acero inoxidable teniendo en cuenta los diseños en Software CAD y se basaron en los cálculos del objetivo dos que dieron las dimensiones del mecanismo, en la parte electrónica se plasman los cálculos hallados, para poder encontrar la trayectoria de cada uno de los pacientes de acuerdo a sus dimensiones, esta parte electrónica consta de una tarjeta puente H para controlar los motores por medio del PWM, una tarjeta Arduino que hace las veces de tarjeta principal, Encoders incrementales para medir el Ángulo, una celda de carga tipo S que se usa para medir la carga en el mecanismo y la fuerza que hace la persona en el robot entre otros electrónicos. También se realizó una comunicación con Arduino mega con otra tarjeta de desarrollo en este caso SP32 la cual es la encargada del HMI interfaz hombre – máquina, SP32 como web Server que se conecta a internet y se accede a ella por página web y desde allí se pueden enviar los datos del paciente y la información que se quiere que el robot tenga para realizar la trayectoria y ejercicios personalizados teniendo en cuenta los ángulos obtenidos.

Para finalizar el desarrollo del prototipo robótico para la rehabilitación de rodilla enfocado al fortalecimiento muscular se realizó con el objetivo de ayudar a los fisioterapeutas a realizar los ejercicios a los diferentes pacientes. Facilitando de esta manera que los profesionales de la salud puedan tener más certeza acerca de los arcos de movilidad y la fuerza que realizan los pacientes, haciendo de esta forma posible llevar con una exactitud mayor estas mediciones. También se le

proporciona a los fisioterapeutas un dispositivo que les permita hacer sus ejercicios de rehabilitación de manera más precisa. Aunque el robot puede llegar a tener cierta autonomía siempre debe operarse bajo la supervisión de un profesional de la salud.

Bibliografía

- [1] B. Ángel, F. César, and M. José, *Tratamiento Fisioterápico de la Rodilla*. MacGraw-Hill, 2003.
- [2] G. Chen, P. Qi, Z. Guo, and H. Yu, “Mechanical design and evaluation of a compact portable knee – ankle – foot robot for gait rehabilitation,” vol. 103, pp. 51–64, 2016.
- [3] A. Beatriz *et al.*, *Rehabilitación en salud Una mirada médica necesaria*, Primera Ed. Medellín: Universidad de Antioquia, 1995.
- [4] A. S. Yusof, A. I. Che-Ani, Z. Hussain, N. Hamzah, R. Boudville, and M. F. A. Rahman, “Back-drivability of powered knee orthosis for knee free swing and knee extension,” *Proc. - 7th IEEE Int. Conf. Control Syst. Comput. Eng. ICCSCE 2017*, vol. 2017-Novem, no. November, pp. 331–335, 2018, doi: 10.1109/ICCSCE.2017.8284429.
- [5] S. W. O’Driscoll and N. J. Giori, “Department of Veterans Affairs Continuous passive motion (CPM) : Theory and principles of clinical application,” *J. Rehabil. Res. Dev.*, vol. 37, no. 2, pp. 179–188, 2000.
- [6] A. Blanco Ortega, H. R. Azcaray Rivera, L. G. Vela Valdés, and R. F. Vasquéz Bautista, “Máquina de Rehabilitación de Tobillo: prototipo virtual y físico,” *X Congr. Int. sobre Innovación y Desarro. Tecnológico CIINDET 2013*, no. March, p. 7, 2013.
- [7] J. Nikitzuk, B. Weinberg, P. K. Canavan, and C. Mavroidis, “Active knee rehabilitation orthotic device with variable damping characteristics implemented via an electrorheological fluid,” *IEEE/ASME Trans. Mechatronics*, vol. 15, no. 6, pp. 952–960, 2010, doi: 10.1109/TMECH.2009.2036170.
- [8] G. Chen, P. Qi, Z. Guo, and H. Yu, “Mechanical design and evaluation of a compact portable knee-ankle-foot robot for gait rehabilitation,” *Mech. Mach. Theory*, vol. 103, pp. 51–64, 2016, doi: 10.1016/j.mechmachtheory.2016.04.012.
- [9] D. Y. Grebnev, I. Y. Maklakova, and A. P. Yastrebov, “The prospect of using stem cells to activate hematopoiesis during age-related involution under radiation exposure,” *Adv. Gerontol.*, vol. 5, no. 2, pp. 80–83, 2015, doi: 10.1016/j.sna.2018.05.016.
- [10] C. H. Guzmán, A. Blanco, J. A. Brizuela, and F. A. Gómez, “Robust control of a hip–joint rehabilitation robot,” *Biomed. Signal Process. Control*, vol. 35, pp. 100–109, 2017, doi:

- 10.1016/j.bspc.2017.03.002.
- [11] Z. Y. Wong, C. M. Teng, and Y. Z. Chong, “Power assisted pneumatic-based knee-ankle-foot-orthosis for rehabilitation,” *2012 IEEE-EMBS Conf. Biomed. Eng. Sci. IECBES 2012*, no. December, pp. 300–304, 2012, doi: 10.1109/IECBES.2012.6498058.
- [12] W. M. dos Santos, G. A. P. Caurin, and A. A. G. Siqueira, “Design and control of an active knee orthosis driven by a rotary Series Elastic Actuator,” *Control Eng. Pract.*, vol. 58, pp. 307–318, 2017, doi: 10.1016/j.conengprac.2015.09.008.
- [13] M. Russo, M. Ceccarelli, and Y. Takeda, *Comparison of motion/force transmissibility in a 3-SPR parallel manipulator and a 6-SPS equivalent mechanism*, vol. 72. Springer International Publishing, 2019.
- [14] L. Cheng, M. Chen, and Z. Li, “Design and Control of a Wearable Hand Rehabilitation Robot,” *IEEE Access*, vol. 6, no. 5, pp. 74039–74050, 2018, doi: 10.1109/ACCESS.2018.2884451.
- [15] J. Araque, C. Peña, and G. Moreno, “Synthesis for a Knee Rehabilitation Mechanism Applying Genetic Algorithms,” *Int. J. Appl. Eng. Res. ISSN 0973-4562 Vol. 13, Number 19 pp. 14451-14456 © Res. India Publ. http://www.ripublication.com*, vol. 13, no. 19, pp. 14451–14456, 2018.
- [16] G. Xu and G. Yu, “Reprint of: On convergence analysis of particle swarm optimization algorithm,” *J. Comput. Appl. Math.*, vol. 340, pp. 709–717, 2018, doi: 10.1016/j.cam.2018.04.036.
- [17] G. Huang *et al.*, “Master-Slave Control of an Intention-Actuated Exoskeletal Robot for Locomotion and Lower Extremity Rehabilitation,” *Int. J. Precis. Eng. Manuf.*, vol. 19, no. 7, pp. 983–991, 2018, doi: 10.1007/s12541-018-0116-x.
- [18] M. Munlin and M. Anantathanavit, “New social-based radius particle swarm optimization,” *Proc. 2017 12th IEEE Conf. Ind. Electron. Appl. ICIEA 2017*, vol. 2018-Febru, pp. 838–843, 2018, doi: 10.1109/ICIEA.2017.8282956.
- [19] M. Anantathanavit and M. A. Munlin, “Radius particle swarm optimization,” *2013 Int. Comput. Sci. Eng. Conf. ICSEC 2013*, pp. 126–130, 2013, doi: 10.1109/ICSEC.2013.6694765.
- [20] J. O. Ao and O. V. Bernartt, “‘ Florian Opolis 2008,” 2008.
- [21] B. Weinberg *et al.*, “Design, control and human testing of an active knee rehabilitation

- orthotic device,” *Proc. - IEEE Int. Conf. Robot. Autom.*, no. April, pp. 4126–4133, 2007, doi: 10.1109/ROBOT.2007.364113.
- [22] E. H. Murai, *Mecanismos Notas de Aula para la Disciplina*, 1st ed. Florianópolis, 2016.
- [23] H. R. Cazangi, “Aplicación de Método de Davies para el Análisis Cinemático y Estático de Mecanismos con Múltiples Grados de Libertad,” Universidad Federal de Santa Catarina, 2008.
- [24] A. P. García and J. L. D. Rodríguez, *Fundamentos en Sistemas de Control Automático*, 1st ed. Pamplona, 2004.

ANEXOS

Anexo 1. Función código trayectorias Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [pxx,pyy,j] = Trayectorias
L1 = 36; % Medida de cadera-rodilla
L2 = 36; % Medida de rodilla-tobillo

CR = [40 45 43.5 50 41 51 33 35 37 39 24]; % Medida de cadera-rodilla
pacientes
RT = [40 45 44 48 38 47 32 43 38 39 17]; % Medida de rodilla-tobillo
pacientes

%Constantes de la curva que da la trayectoria
p1 = -793.4;
p2 = -3.047e+05;
p3 = 1.36e+04;
q1 = -1.981e+04;
q2 = 6.556e+04;

x = 0; %linspace(0,-45,50);%valores en el eje x
i = 1;
j = 1;
for j = 1:size(CR,2)
    i = 1;
    for x = 0:-1:-50
        y = ((p1*(x^2))+(p2*x)+p3)/((x^2)+(q1*x)+q2); %ecuacion que describe
la trayectoria
        px(i) = x;
        py(i) = y;
        [theta,alpha,beta] = Tray_Dir(L1,L2,x,y);
        [x1,y1] = Tray_Inv(CR(j),RT(j),theta,alpha,beta);
        AM(j,i) = 180-(beta*(180/pi)); %Arco de movilidad
        pxx(j,i) = x1;
        pyy(j,i) = y1;
        i = i+1;
    end
end
ta = theta + alpha;
x2 = L1*cos(ta)-(L1+L2);
y2 = L1*sin(ta);

% plot(px,py)
for s = 1:j
    hold on
    plot(pxx(s,:),pyy(s,:))
end
```

Anexo 2. Función transformar trayectorias Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function trayectoria = transformar_trayectoria(Dx, Dy, Rz, pxx, pyy, w);

%Constantes de la curva que da la trayectoria
p1 = -793.4;
p2 = -3.047e+05;
p3 = 1.36e+04;
q1 = -1.981e+04;
q2 = 6.556e+04;

Px = linspace(0,-45,50);%valores en el eje x
Py = ((p1*(Px.^2))+p2*Px+p3)./((Px.^2)+(q1*Px)+q2);%ecuacion que describe
la trayectoria
A = MTHtrasy(Dy)*MTHtrasx(Dx)*MTHRotz(Rz*pi/180);%Matrices de transformacion
homogenea y los valores dados a trasladar
%[pxx,pyy] = Trayectorias;
P = [pxx(w,:) ; pyy(w,:) ; zeros(1,length(pxx)) ; ones(1,length(pxx))];%[pxx ;
pyy; zeros(1,length(pxx)) ; ones(1,length(pyy))];% %matriz de los puntos que
describen la trayectoria
trayectoria= A*P;%punto trasladado
```

Anexo 3. Trayectorias tobillo Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, close all, clc

L1 = 36; % Medida de cadera-rodilla
L2 = 36; % Medida de rodilla-tobillo

CR = [36 40 45 43.5 50 41 51 33 35 37 39 24]; % Medida de cadera-rodilla
pacientes
RT = [36 40 45 44 48 38 47 32 43 38 39 17]; % Medida de rodilla-tobillo
pacientes

%Constantes de la curva que da la trayectoria
p1 = -793.4;
p2 = -3.047e+05;
p3 = 1.36e+04;
q1 = -1.981e+04;
q2 = 6.556e+04;

x = 0; %linspace(0,-45,50);%valores en el eje x
i = 1;
j = 1;
for j = 1:5%size(CR,2)
    i = 1;
    for x = 0:-1:-50
        y = ((p1*(x^2)))+(p2*x)+p3)/((x^2)+(q1*x)+q2); %ecuacion que describe
la trayectoria
        px(i) = x;
        py(i) = y;
        [theta,alpha,beta] = Tray_Dir(L1,L2,x,y);
        [x1,y1] = Tray_Inv(CR(j),RT(j),theta,alpha,beta);
        pxx(j,i) = x1;
        pyy(j,i) = y1;
        AM(j,i) = 180-(beta*(180/pi)); %Arco de movilidad
        i = i+1;
    end
end
end
pxx
pyy
ta = theta + alpha;
x2 = L1*cos(ta)-(L1+L2);
y2 = L1*sin(ta);

plot(px,py,'LineWidth',3)
for s = 1:j
    hold on
    plot(pxx(s,:),pyy(s,:), 'LineWidth',2)
```

```
end
z = legend('Patient 0','Patient 1','Patient 2','Patient 3','Patient
4','Location','southwest');%,'Patient 5','Patient 6','Patient 7','Patient
8','Patient 9','Patient 10','Patient 11','Location','southwest');
set(z,'FontName','Times New Roman','FontSize', 11);
xlabel('X Ankle Position','FontName','Times New Roman','FontSize', 11);
ylabel('Y Ankle Position','FontName','Times New Roman','FontSize', 11);
```

Anexo 4. Función trayectoria directa Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [theta,alpha,beta] = Tray_Dir(L1,L2,x,y)

x = x*(-1);
if x == 0
    theta = 0;
    alpha = 0;
    beta = pi;
else
    C = L1+L2-x;
    J = sqrt((C^2)+(y^2));
    st = y/J;
    ct = C/J;
    theta = atan2(st,ct);
    ca = ((L1^2)+(J^2)-(L2^2))/(2*L1*J);
    sa = 1*sqrt(1-(ca^2));
    alpha = atan2(sa,ca);
    cb = ((L1^2)+(L2^2)-(J^2))/(2*L1*L2);
    sb = 1*sqrt(1-(cb^2));
    beta = atan2(sb,cb);
end
```

Anexo 5. Función trayectoria inversa Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [x,y] = Tray_Inv(L1,L2,theta,alpha,beta)

if beta == pi
    x = 0;
    y = 0;
else
    J = sqrt((L1^2)+(L2^2)-(2*L1*L2*cos(beta)));
    y = J*sin(theta);
    C = J*cos(theta);
    x = L1+L2-C;
    x = x*(-1);
end
```


Anexo 6. Función cinemática directa mecanismo sin pierna Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Robot_dibujo = cin_dir_5b_2(dim_robot,t1,t2,tp1,tp2)

% dim_robot = [L, L1, L2, Dx, Dy, Rz];

%dimensiones del robot
L = dim_robot(1);
L1 = dim_robot(2);
L2 = dim_robot(3);
L3 = dim_robot(2);
L4 = dim_robot(3);
Lp1 = 36;
Lp2 = 36;

F = Lp1+Lp2;
% pd = [-p,0];
% pil = [-L-p,0];
tetta(1) = t1;
tetta(2) = t2;

% Punto de origen.
Ax=0;
Ay=0;

% Ubicación del puento B.
Bx=L1*cos(tetta(1))+Ax;
By=L1*sin(tetta(1))+Ay;

% Ubicación del puento C.
Cx=L+Ax;
Cy=Ay;

% Ubicación del puento D.
Dx=(L3*cos(tetta(2)))+L+Ax;
Dy=L3*sin(tetta(2))+Ay;

% Barra imaginaria R.
R=sqrt((L-L1*cos(tetta(1))+L3*cos(tetta(2)))^2+(L1*sin(tetta(1))-L3*sin(tetta(2)))^2);
tettaR=atan2(L3*sin(tetta(2))-L1*sin(tetta(1)),L-L1*cos(tetta(1))+L3*cos(tetta(2)));

% Cálculo de tetta2.
tetta2=tettaR+acos((R^2+L2^2-L4^2)/(2*R*L2));
```

```

% Punto P (Efector Final).
Px=Bx+L2*cos(tettal2);
Py=By+L2*sin(tettal2);

tpt = tp1 + tp2;
ppx0 = -F;
ppy0 = 0;
ppx1 = ppx0 + Lp1*cos(tpt);
ppy1 = Lp1*sin(tpt);
% Almacenamiento de las variables de interés en una estructura.
Robot_dibujo.Ax=Ax;
Robot_dibujo.Ay=Ay;
Robot_dibujo.Bx=Bx;
Robot_dibujo.By=By;
Robot_dibujo.Cx=Cx;
Robot_dibujo.Cy=Cy;
Robot_dibujo.Dx=Dx;
Robot_dibujo.Dy=Dy;
Robot_dibujo.Px=Px;
Robot_dibujo.Py=Py;
Robot_dibujo.ppx0=ppx0;
Robot_dibujo.ppy0=ppy0;
% Robot_dibujo.ppx1=ppx1;
% Robot_dibujo.ppy1=ppy1;

end

```

Anexo 7. Función cinemática inversa mecanismo sin pierna Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [t1,t2,v,tp1,tp2] = cin_inv_Paralelo_5b(dim_robot,px,py)

L = dim_robot(1,1);
L1 = dim_robot(1,2);
L2 = dim_robot(1,3);
L3 = L1;
L4 = L2;
Lp1 = 36;
Lp2 = 36;
pd = [L,0];
pi1 = [0,0];
pob = [px,py];
a = norm(pob-pi1);
b = norm(pob-pd);
if a >= L1+L2
    t1 = 0;
    t2 = 0;
    tp1 = 0;
    tp2 = 0;
    v = 0;
    return
end
if b >= L3+L4
    t1 = 0;
    t2 = 0;
    tp1 = 0;
    tp2 = 0;
    v = 0;
    return
end
% t1 = 1;
% t2 = 2;
% v = 1;
CODO1=1;
CODO2=1;

x = px;
y = py;

% a=sqrt(x^2 + y^2);
% b=sqrt((L-x)^2 + y^2);

alpha=atan2(y,x);%%%
%omega=pi-atan2(y,L-x);
```

```

omega=atan2(y,x-L);

cosphi=(L1^2+L2^2-a^2)/(2*L1*L2);
senphi=CODO1*sqrt(1-cosphi^2);

if not(isreal(cosphi)) | not(isreal(senphi))
    t1 = 0;
    t2 = 0;
    tp1 = 0;
    tp2 = 0;
    v = 0;
    return
end

phi=atan2(senphi,cosphi);

cospsi=(L3^2+L4^2-b^2)/(2*L3*L4);
senpsi=CODO2*sqrt(1-cospsi^2);

if not(isreal(cospsi)) | not(isreal(senpsi))
    t1 = 0;
    t2 = 0;
    tp1 = 0;
    tp2 = 0;
    v = 0;
    return
end
psi=atan2(senpsi,cospsi);

betta=atan2(L2*sin(pi-phi),L1+L2*cos(pi-phi));
sigma=atan2(L4*sin(pi-psi),L3+L4*cos(pi-psi));

% Posiciones angulares de interés.
t1=alpha+betta;
t2=omega-sigma;

if t1 > 270*pi/180 || t1 < 30*pi/180 || t2 > 150*pi/180 || t2 < -90*pi/180 %
VERIFICAR CONDICION
% if t1 > 3*pi/2 || t2 > pi
    v = 0;
else
    v = 1;
end

F = Lp1+Lp2;
h = y;
xp = F + x;
tp1 = atan2(h,xp);
jp = sqrt((xp^2)+(h^2));
tp2 = acos(((Lp1^2)+(jp^2)-(Lp2^2))/(2*Lp1*jp));

```

Anexo 8. Función dibujar robot mecanismo sin pierna Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function dibujar_robot_5b_Ap(Robot_dibujo,A,A1)

escala=1;

% Configuración de la figura.
axis([-100*escala 10*escala -10*escala 50*escala -50*escala 50*escala]);
hold on

% set(gcf,'color','w');
% set(gca,'xcolor','W')
% set(gca,'ycolor','W')
% set(gca,'zcolor','W')
% set(gca,'color','w')
%

view(-120,60);

% Extracción de datos.
Ax=A*Robot_dibujo.Ax;
Ay=A*Robot_dibujo.Ay;
Bx=A*Robot_dibujo.Bx;
By=A*Robot_dibujo.By;
Cx=A*Robot_dibujo.Cx;
Cy=A*Robot_dibujo.Cy;
Dx=A*Robot_dibujo.Dx;
Dy=A*Robot_dibujo.Dy;
Px=A*Robot_dibujo.Px;
Py=A*Robot_dibujo.Py;
% ppx0=Robot_dibujo.ppx0;
% ppy0=Robot_dibujo.ppy0;
% ppx1=Robot_dibujo.ppx1;
% ppy1=Robot_dibujo.ppy1;

%se traslada cada punto del robot a graficar
z=50;
Axy=A*[Robot_dibujo.Ax;Robot_dibujo.Ay; z; 1];
Ax=Axy(1);, Ay=Axy(2);
Bxy=A*[Robot_dibujo.Bx;Robot_dibujo.By; z; 1];
Bx=Bxy(1);, By=Bxy(2);
Cxy=A*[Robot_dibujo.Cx;Robot_dibujo.Cy; z; 1];
Cx=Cxy(1);, Cy=Cxy(2);
Dxy=A*[Robot_dibujo.Dx;Robot_dibujo.Dy; z; 1];
Dx=Dxy(1);, Dy=Dxy(2);
Pxy=A*[Robot_dibujo.Px;Robot_dibujo.Py; z; 1];
```

```

Px=Pxy(1);, Py=Pxy(2);
% ppxyl=A1*[Robot_dibujo.Px;Robot_dibujo.Py; z; 1];
% ppxl=ppxy1(1);, ppy1=ppxy1(2);

% Dibujo de los soportes fijos (Hombros).
% h1=30*escala; h2=20*escala;
% Sopx=[Ax Ax-h2 Ax+h2 Ax];
% Sopy=[Ay Ay-h1 Ay-h1 Ay];
% plot3(Sopx,Sopy,[z,z,z,z],'r');
% Sopx=[Cx Cx-h2 Cx+h2 Cx];
% Sopy=[Cy Cy-h1 Cy-h1 Cy];
% plot3(Sopx,Sopy,[z,z,z,z],'r');

% Dibujo de puntos.
plot3(Ax,Ay,z,'or','LineWidth',3);
plot3(Bx,By,z,'ob','LineWidth',3);
plot3(Cx,Cy,z,'or','LineWidth',3);
plot3(Dx,Dy,z,'ob','LineWidth',3);
plot3(Px,Py,z,'ob','LineWidth',3);

% Nombre de los puntos.
% text(Ax-(0.5*escala),Ay-(2.6*escala),z,'A','color','b');
% text(Bx-(1.2*escala),By+(1.5*escala),z,'B','color','b');
% text(Cx-(0.5*escala),Cy-(2.6*escala),z,'C','color','b');
% text(Dx+(0.6*escala),Dy+(1.5*escala),z,'D','color','b');
% text(Px-(0.5*escala),Py+(1.4*escala),z,'P','color','b');

% Dibujo de barras.
plot3([Ax Cx],[Ay Cy],[z,z],'-b','LineWidth',0.5); % Barra L.
plot3([Ax Bx],[Ay By],[z,z],'k','LineWidth',8); % Barra L1.
plot3([Cx Dx],[Cy Dy],[z,z],'k','LineWidth',8); % Barra L3.
plot3([Bx Px],[By Py],[z,z],'k','LineWidth',8); % Barra L2.
plot3([Dx Px],[Dy Py],[z,z],'k','LineWidth',8); % Barra L4.
% plot3([ppx0 ppx1],[ppy0 ppy1],[z,z],'g','LineWidth',8);
% plot3([ppx1 Px],[ppy1 Py],[z,z],'g','LineWidth',8);

% Nombre de las barras.
% text((Cx-Ax)/2-0.5,-2.6*escala,z,'L','color','k');
% text(Bx/2-2.2,By/2,z,'L1','color','k');
% text((Cx-Ax)+(Dx-Cx+Ax)/2+0.8*escala,Dy/2,z,'L3','color','k');
% text((Px+Bx)/2-1.5*escala,By+(Py-By)/2+1.5*escala,z,'L2','color','k');
% text(Px+(Dx-Px)/2+0.4*escala,Dy+(Py-Dy)/2+1.5*escala,z,'L4','color','k');

% Sistema coordenado.
% plot3([Ax Ax],[-2*escala+Ay 2*escala+Ay],[z,z],'g','LineWidth',3*escala);
% plot3([-2*escala+Ax 2*escala+Ax],[Ay,Ay],[z,z],'r','LineWidth',3*escala);
% plot3([Ax Ax],[Ay,Ay],[-2*escala+z 2*escala+z],'b','LineWidth',3*escala);

```

```
% text(50*escala,Ay+8*escala,z,'x'); % Etiqueta eje x.
% text(Ax+8*escala,50*escala,z,'y'); % Etiqueta eje y.
% text(Ax+8*escala,Ay,z+50*escala,'z'); % Etiqueta eje z.

% %% Comentarios.
% text(-25*escala,-12*escala,'Desarrollado por: Efraín A. Rodríguez
Gasca','color','b');
% text(-11.5*escala,-14*escala,'Eliseo Cortés Torres','color','b');
% text(-25*escala,-16.5*escala,'Ingeniería Mecatrónica','color','b');
% text(-25*escala,-19*escala,'UNIVERSIDAD DE PAMPLONA','color','b');

end
```

Anexo 9. Función matriz de transformación homogénea rotación en Z mecanismo sin pierna Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function MTH = MTHRotz(angulo)

MTH = [cos(angulo), -sin(angulo), 0 ,0 ;...
sin(angulo), cos(angulo), 0 ,0 ;...
0, 0, 1 ,0 ;...
0, 0, 0 ,1];
```


Anexo 10. Función matriz de transformación homogénea traslación en X mecanismo sin pierna Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function MTH = MTHtrasx(distancia)

MTH = eye(4);
MTH(1,4) = distancia;
```

Anexo 11. Función matriz de transformación homogénea traslación en Y mecanismo sin pierna Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function MTH = MTHtrasy(distancia)

MTH = eye(4);
MTH(2,4) = distancia;
```

Anexo 12. Optimización mecanismo sin pierna Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear, close all, clc

%% MAXIMOS Y MINIMOS DE LA FUNCION FITNESS
MaxL = 50;
MinL = 10;
MaxL1 = 50;
MinL1 = 5;
MaxL2 = 50;
MinL2 = 5;
MaxDx = 50;
MinDx = -50;
MaxDy = 50;
MinDy = -50;
% MaxRz = 20;
% MinRz = -20;

%% SE OPTIMIZA LAS DIMENSIONES DEL ROBOT POR MEDIO DE EJAMBRE DE PARTICULAS
FitnessFunction = @verificar_robot_y_trayectoria_sinRz3; %se asigna a una
variable la funcion verificar
numberOfVariables = 5; %se asigna el numero de variables a optimizar
rng default

[x, fval, exitflag, output] = particleswarm(FitnessFunction,...
    numberOfVariables, [MinL, MinL1, MinL2, MinDx, MinDy],...
    [MaxL, MaxL1, MaxL2, MaxDx, MaxDy] );%se usa la funcion de optimizacion

figure(2);, cla;
plot(x, fval, 'x');%se dibuja el robot con sus dimensiones optimizadas

x
```

Anexo 13. Función verificación mecanismo sin pierna Matlab

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f=verificar_robot_y_trayectoria_sinRz3(dim_robot)

ANIMAR = 0;%variable que define si se grafica cada una de las opciones a
graficar
dim_robot=[dim_robot, 0];%variable de las dimensiones del robot
[pxx,pyy,j] = Trayectorias;
%trayectoria = transformar_trayectoria2(dim_robot(4), dim_robot(5),
dim_robot(6));%se genera la trayectoria a seguir

% verificacion si el robot es capaz de realizar toda la trayectoria
for w = 1:j
    trayectoria = transformar_trayectoria2(dim_robot(4), dim_robot(5),
dim_robot(6), pxx, pyy, w);%se genera la trayectoria a seguir
    for i = 1:length(trayectoria)
        i;
        [t1,t2,v,tp1,tp2] =
cin_inv_Paralelo_5b(dim_robot,trayectoria(1,i),trayectoria(2,i));%funcion de
cinematica inversa
        if v==0 %si v es cero quiere decir que alguno de los puntos de la
trayectoria no lo pudo alcanzar el robot
            f = 10e10;%se coloca un error grande cuando las dimensiones no son
adecuadas
            return;
        end
        if ANIMAR==1
            figure(1);, cla;
            Robot_dibujo = cin_dir_5b_2(dim_robot,t1,t2,tp1,tp2);
            dibujar_robot_5b(Robot_dibujo);%se dibuja el robot
            plot (trayectoria(1,:),trayectoria(2,:))%se dibuja la trayectoria
            view(0,90);
            pause(0.1);
        end
        if ANIMAR==2
            A =
MTHtrasy(dim_robot(5))*MTHtrasx(dim_robot(4))*MTHRotz(dim_robot(6)*pi/180);%s
e traslada la posicion del robot
            A1 = MTHtrasy(0)*MTHtrasx(0);
            trayectoria_org = transformar_trayectoria(0,0,0);%se deja la
trayectoria original
            figure(1);, cla;
            Robot_dibujo = cin_dir_5b_2(dim_robot,t1,t2,tp1,tp2);
            %dibujar_robot_5b_A(Robot_dibujo,inv(A));%se dibuja el robot
            dibujar_robot_5b_Ap(Robot_dibujo,inv(A),inv(A1));%se dibuja el robot
            plot (trayectoria_org(1,:),trayectoria_org(2:,:), '-
b', 'LineWidth',3)%se dibuja la trayectoria
            view(0,90);

```

```
        pause(0.1);
        grid on
    end
end
end

f = dim_robot(1)+(2*dim_robot(2))+(2*dim_robot(3));%funcion a optimizar
```

Anexo 14. Función cinemática directa mecanismo con pierna Matlab

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Robot_dibujo = cin_dir_5b_pi(dim_robot,t1,t2)%,tp1,tp2)

% dim_robot = [5.0000    8.4    18.9    6    -2];
%
% t1 = 110;
% t2 = 60;
% a = 45;
L = dim_robot(1);
L1 = dim_robot(2);
L2 = dim_robot(3);
L3 = dim_robot(2);
L4 = dim_robot(3);
L5 = dim_robot(6);
alpha = dim_robot(7)*pi/180;
tetta(1) = t1;
tetta(2) = t2;

% Punto de origen.
Ax=0;
Ay=0;

% Ubicación del puento B.
Bx=L1*cos(tetta(1))+Ax;
By=L1*sin(tetta(1))+Ay;

% Ubicación del puento C.
Cx=L+Ax;
Cy=Ay;

% Ubicación del puento D.
Dx=(L3*cos(tetta(2)))+L+Ax;
Dy=L3*sin(tetta(2))+Ay;

% Barra imaginaria R.
R=sqrt((L-L1*cos(tetta(1))+L3*cos(tetta(2)))^2+(-L1*sin(tetta(1))+L3*sin(tetta(2)))^2);
tettaR=atan2(L3*sin(tetta(2))-L1*sin(tetta(1)),L-L1*cos(tetta(1))+L3*cos(tetta(2)));

% Cálculo de tetta2.
tetta2=tettaR+acos((R^2+L2^2-L4^2)/(2*R*L2));

% Punto P.
Px=Bx+L2*cos(tetta2);
Py=By+L2*sin(tetta2);

```

```
% Efecto final F
Fx=Bx+L5*cos(tettal2+alpha);
Fy=By+L5*sin(tettal2+alpha);

% Almacenamiento de las variables de interés en una estructura.
Robot_dibujo.Ax=Ax;
Robot_dibujo.Ay=Ay;
Robot_dibujo.Bx=Bx;
Robot_dibujo.By=By;
Robot_dibujo.Cx=Cx;
Robot_dibujo.Cy=Cy;
Robot_dibujo.Dx=Dx;
Robot_dibujo.Dy=Dy;
Robot_dibujo.Px=Px;
Robot_dibujo.Py=Py;
Robot_dibujo.Fx=Fx;
Robot_dibujo.Fy=Fy;
```

Anexo 15. Función cinemática inversa mecanismo con pierna Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function [t11,t22,v] = cin_inv_5_pi221(dim_robot,Fx,Fy)

L = dim_robot(1);
L1 = dim_robot(2);
L2 = dim_robot(3);
L3 = dim_robot(2);
L4 = dim_robot(3);
L5 = dim_robot(6);
omega = dim_robot(7)*pi/180;
v=0;
t=0;
COD01=1;
COD02=1;
Ax=0;
Ay=0;
Cx=L;
Cy=0;
A = [Ax Ay];
C = [Cx Cy];
F = [Fx Fy];%[0 14.4];
Fax=[0];
Fay=[0];
i=1;
a = norm(F-A);
if a > L1+L5
    t11 = 0;
    t22 = 0;
    v = 0;
    t=1
    %warning('magnitud del punto f es mayor que L1+L5');
    return
end
cosalpha=((a^2)-(L1^2)-(L5^2))/(2*L1*L5);
senalpha=COD01*sqrt(1-cosalpha^2);
if not(isreal(cosalpha)) | not(isreal(senalpha))
    t11 = 0;
    t22 = 0;
    t=2
    v = 0;
    %warning('alpha imaginario');
    return
end
alpha=atan2(senalpha,cosalpha);
x15=L5*cos(alpha);
y15=L5*sin(alpha);
beta=atan2(y15,x15+L1);
```



```

t11=beta+(atan2(Fy,Fx));
x12=L2*cos(alpha+omega);%L2*sin(alpha+omega);
y12=L2*sin(alpha+omega);%L2*cos(alpha+omega);
ac=sqrt((L2*cos(alpha+omega)+L1)^2 + (L2*sin(alpha+omega))^2);
lamda=atan2(L2*sin(alpha+omega),L2*cos(alpha+omega)+L1);
yota=t11-lamda;
Px=ac*cos(yota);
Py=ac*sin(yota);
P=[Px,Py];
b=norm(P-C);
if b>(L3+L4)
    t=8
    COD01=COD01*(-1);
    cosalpha=((a^2)-(L1^2)-(L5^2))/(2*L1*L5);
    senalpha=COD01*sqrt(1-cosalpha^2);
    alpha=atan2(senalpha,cosalpha);
    x15=L5*cos(alpha);
    y15=L5*sin(alpha);
    beta=atan2(y15,x15+L1);
    t11=beta+(atan2(Fy,Fx));
    x12=L2*cos(alpha+omega);%L2*sin(alpha+omega);
    y12=L2*sin(alpha+omega);%L2*cos(alpha+omega);
    ac=sqrt((L2*cos(alpha+omega)+L1)^2 + (L2*sin(alpha+omega))^2);
    lamda=atan2(L2*sin(alpha+omega),L2*cos(alpha+omega)+L1);
    yota=t11-lamda;
    Px=ac*cos(yota);
    Py=ac*sin(yota);
    P=[Px,Py];
    b=norm(P-C);
end
if b>(L3+L4)
    t11 = 0;
    t22 = 0;
    v = 0;
    t=3
    %warning('magnitud del punto p es mayor que L3+L4');
    return
end
cosphi=((L3^2)+(L4^2)-(b^2))/(2*L3*L4);
sinphi=COD02*sqrt(1-cosphi^2);
if not(isreal(sinphi))
    t=9
    COD02=COD02*(-1);
    cosalpha=((a^2)-(L1^2)-(L5^2))/(2*L1*L5);
    senalpha=COD01*sqrt(1-cosalpha^2);
    if not(isreal(cosalpha)) | not(isreal(senalpha))
        t11 = 0;
        t22 = 0;
        t=4
        v = 0;
        %warning('alpha imaginario');
        return
    end
alpha=atan2(senalpha,cosalpha);
x15=L5*cos(alpha);
y15=L5*sin(alpha);
beta=atan2(y15,x15+L1);

```

```

t11=beta+(atan2(Fy,Fx));
x12=L2*cos(alpha+omega);%L2*sin(alpha+omega);
y12=L2*sin(alpha+omega);%L2*cos(alpha+omega);
ac=sqrt((L2*cos(alpha+omega)+L1)^2 + (L2*sin(alpha+omega))^2);
lamda=atan2(L2*sin(alpha+omega),L2*cos(alpha+omega)+L1);
yota=t11-lamda;
Px=ac*cos(yota);
Py=ac*sin(yota);
P=[Px,Py];
b=norm(P-C);
if b>(L3+L4)
    t=10
    COD01=COD01*(-1);
    cosalpha=((a^2)-(L1^2)-(L5^2))/(2*L1*L5);
    senalpha=COD01*sqrt(1-cosalpha^2);
    alpha=atan2(senalpha,cosalpha);
    x15=L5*cos(alpha);
    y15=L5*sin(alpha);
    beta=atan2(y15,x15+L1);
    t11=beta+(atan2(Fy,Fx));
    x12=L2*cos(alpha+omega);%L2*sin(alpha+omega);
    y12=L2*sin(alpha+omega);%L2*cos(alpha+omega);
    ac=sqrt((L2*cos(alpha+omega)+L1)^2 + (L2*sin(alpha+omega))^2);
    lamda=atan2(L2*sin(alpha+omega),L2*cos(alpha+omega)+L1);
    yota=t11-lamda;
    Px=ac*cos(yota);
    Py=ac*sin(yota);
    P=[Px,Py];
    b=norm(P-C);
end
if b>(L3+L4)
    t11 = 0;
    t22 = 0;
    v = 0;
    t=5
    %warning('magnitud del punto p es mayor que L3+L4');
    return
end
cosphi=((L3^2)+(L4^2)-(b^2))/(2*L3*L4);
sinphi=COD02*sqrt(1-cosphi^2);
end
if not(isreal(cosphi)) | not(isreal(sinphi))
    t11 = 0;
    t22 = 0;
    v = 0;
    t=6
    P;
    L3;
    L4;
    b;
    Fax(i)=Fx;
    Fay(i)=Fy;
    i=i+1;
    %warning('phi imaginario');
    return
end
phi=atan2(sinphi,cosphi);

```

```

cospsi=(b^2)-(L3^2)-(L4^2)/(2*L3*L4);
sinpsi=sqrt(1-cospsi^2);
if not(isreal(cospsi)) | not(isreal(sinpsi))
    t11 = 0;
    t22 = 0;
    v = 0;
    t=7
    %warning('psi imaginario');
    return
end
psi=atan2(sinpsi,cospsi);
beta1=atan2(L4*sin(psi),L4*cos(psi)+L3);
t22=atan2((Py-Cy),(Px-Cx))-beta1;
% if t11 > 180*pi/180 || t11 < 10*pi/180 || t22 > 150*pi/180 || t22 <
10*pi/180 % VERIFICAR CONDICION
% % if t1 > 3*pi/2 || t2 > pi
%     t11 = 0;
%     t22 = 0;
%     %v = 0;
%     t=6;
%     warning('angulo erroneo');
%     return
% end
%t
v = 1;

```



```

Px=Pxy(1);, Py=Pxy(2);
Fxy=A*[Robot_dibujo.Fx;Robot_dibujo.Fy; z; 1];
Fx=Fxy(1);, Fy=Fxy(2);
% Dibujo de los soportes fijos (Hombros).
% h1=30*escala; h2=20*escala;
% Sopx=[Ax Ax-h2 Ax+h2 Ax];
% Sopy=[Ay Ay-h1 Ay-h1 Ay];
% plot3(Sopx,Sopy,[z,z,z,z],'r');
% Sopx=[Cx Cx-h2 Cx+h2 Cx];
% Sopy=[Cy Cy-h1 Cy-h1 Cy];
% plot3(Sopx,Sopy,[z,z,z,z],'r');

% Dibujo de puntos.
plot3(Ax,Ay,z,'or');
plot3(Bx,By,z,'ob');
plot3(Cx,Cy,z,'or');
plot3(Dx,Dy,z,'ob');
plot3(Px,Py,z,'ob');
plot3(Fx,Fy,z,'ob');

% Nombre de los puntos.
% text(Ax-(0.5*escala),Ay-(2.6*escala),z,'A','color','b');
% text(Bx-(1.2*escala),By+(1.5*escala),z,'B','color','b');
% text(Cx-(0.5*escala),Cy-(2.6*escala),z,'C','color','b');
% text(Dx+(0.6*escala),Dy+(1.5*escala),z,'D','color','b');
% text(Px-(0.5*escala),Py+(1.4*escala),z,'P','color','b');

% Dibujo de barras.
plot3([Ax Cx],[Ay Cy],[z,z],'-.b','LineWidth',0.1); % Barra L.
plot3([Ax Bx],[Ay By],[z,z],'k','LineWidth',5); % Barra L1.
plot3([Cx Dx],[Cy Dy],[z,z],'k','LineWidth',5); % Barra L3.
plot3([Bx Px],[By Py],[z,z],'k','LineWidth',5); % Barra L2.
plot3([Dx Px],[Dy Py],[z,z],'k','LineWidth',5); % Barra L4.
plot3([Bx Fx],[By Fy],[z,z],'k','LineWidth',5); % Barra L5.
plot3([Fx Px],[Fy Py],[z,z],'k','LineWidth',5); % Barra L5.

% Nombre de las barras.
% text((Cx-Ax)/2-0.5,-2.6*escala,z,'L','color','k');
% text(Bx/2-2.2,By/2,z,'L1','color','k');
% text((Cx-Ax)+(Dx-Cx+Ax)/2+0.8*escala,Dy/2,z,'L3','color','k');
% text((Px+Bx)/2-1.5*escala,By+(Py-By)/2+1.5*escala,z,'L2','color','k');
% text(Px+(Dx-Px)/2+0.4*escala,Dy+(Py-Dy)/2+1.5*escala,z,'L4','color','k');

% Sistema coordinado.
% plot3([Ax Ax],[-10*escala+Ay 10*escala+Ay],[z,z],'g','LineWidth',3*escala);
% plot3([-10*escala+Ax 10*escala+Ax],[Ay,Ay],[z,z],'r','LineWidth',3*escala);
% plot3([Ax Ax],[Ay,Ay],[-10*escala+z 10*escala+z],'b','LineWidth',3*escala);
%
% text(50*escala,Ay+8*escala,z,'x'); % Etiqueta eje x.
% text(Ax+8*escala,50*escala,z,'y'); % Etiqueta eje y.
% text(Ax+8*escala,Ay,z+50*escala,'z'); % Etiqueta eje z.

```

```
% Dibujo de la estructura.
% switch unid
%     case 'mm'
%
%     case 'in'
%
%     otherwise
%         %...
% end

% %% Comentarios.
% text(-25*escala,-12*escala,'Desarrollado por: Efraín A. Rodríguez
Gasca','color','b');
% text(-11.5*escala,-14*escala,'Eliseo Cortés Torres','color','b');
% text(-25*escala,-16.5*escala,'Ingeniería Mecatrónica','color','b');
% text(-25*escala,-19*escala,'UNIVERSIDAD DE PAMPLONA','color','b');

end
```

Anexo 17. Optimización mecanismo con pierna Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear, close all, clc

%% MAXIMOS Y MINIMOS DE LA FUNCION FITNESS
MaxL = 50;
MinL = 5;
MaxL1 = 50;
MinL1 = 5;
MaxL2 = 50;
MinL2 = 5;
MaxL5 = 50;%
MinL5 = 5;%
MaxDx = 50;
MinDx = -50;
MaxDy = 50;
MinDy = 0;%-50;
% MaxRz = 20;
% MinRz = -20;

%% SE OPTIMIZA LAS DIMENSIONES DEL ROBOT POR MEDIO DE EJAMBRE DE PARTICULAS
FitnessFunction = @verificar_robot_y_trayectoria_sinRz3; %se asigna a una
variable la funcion verificar
numberOfVariables = 6; %se asigna el numero de variables a optimizar
rng default

[x, fval, exitflag, output] = particleswarm(FitnessFunction,...
    numberOfVariables, [MinL, MinL1, MinL2, MinDx, MinDy, MinL5],...
    [MaxL, MaxL1, MaxL2, MaxDx, MaxDy, MaxL5] );%se usa la funcion de
optimizacion

figure(2);, cla;
plot(x, fval, 'x');%se dibuja el robot con sus dimensiones optimizadas

x
```

Anexo 18.Función verificación mecanismo con pierna Matlab

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function f=verificar_robot_y_trayectoria_sinRz3(dim_robot)

ANIMAR = 0;%variable que define si se grafica cada una de las opciones a
graficar
a = 40;
dim_robot=[10.0004 17.3687 25.4656 dim_robot(4) dim_robot(5) dim_robot(6) a
0];%variable de las dimensiones del robot
[pxx,pyy,j] = Trayectorias;
%trayectoria = transformar_trayectoria2(dim_robot(4), dim_robot(5),
dim_robot(6));%se genera la trayectoria a seguir

% verificacion si el robot es capaz de realizar toda la trayectoria
for w = 1:j
    trayectoria = transformar_trayectoria2(dim_robot(4), dim_robot(5),
dim_robot(8), pxx, pyy, w);%se genera la trayectoria a seguir
    for i = 1:length(trayectoria)
        i;
        [t1,t2,v] =
cin_inv_5_pi221(dim_robot,trayectoria(1,i),trayectoria(2,i));%funcion de
cinematica inverssa
        if v==0 %si v es cero quiere decir que alguno de los puntos de la
trayectoria no lo pudo alcanzar el robot
            f = 10e10;%se coloca un error grande cuando las dimensiones no son
adecuadas
            return;
        end
        if ANIMAR==1
            figure(1);, cla;
            Robot_dibujo = cin_dir_5_pi(dim_robot,t1,t2);
            dibujar_robot_5b_pi(Robot_dibujo);%se dibuja el robot
            plot (trayectoria(1,:),trayectoria(2,:))%se dibuja la trayectoria
            view(0,90);
            pause(0.1);
        end
        if ANIMAR==2
            A =
MTHtrasy(dim_robot(5))*MTHtrasx(dim_robot(4))*MTHRotz(dim_robot(8)*pi/180);%s
e traslada la posicion del robot
            A1 = MTHtrasy(0)*MTHtrasx(0);
            trayectoria_org = transformar_trayectoria(0,0,0);%se deja la
trayectoria original
            figure(1);, cla;
            Robot_dibujo = cin_dir_5_pi(dim_robot,t1,t2);
            %dibujar_robot_5b_A(Robot_dibujo,inv(A));%se dibuja el robot
            dibujar_robot_5b_pi(Robot_dibujo,inv(A),inv(A1));%se dibuja el robot

```



```
        plot (trayectoria_org(1,:),trayectoria_org(2,:), '-
b', 'LineWidth',3)%se dibuja la trayectoria
        view(0,90);
        pause(0.1);
        grid on
    end
end
end

%f = dim_robot(1)+(4*dim_robot(2))+(2*dim_robot(3))+(4*dim_robot(6));%funcion
a optimizar
f = dim_robot(6);
```

Anexo 19. Método de Davies mecanismo sin pierna Matlab

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, close all, clc
syms ax ay bx by cx cy ex ey dx dy wa wb wc we wd Rax Ray Rbx Rby Rcx Rcy Rex
Rey Rdx Rdy Ta Td Tc
%% Etapa 1 a,b y c
n=5; %numero de barras
e=5; %numero de acoples
lambda=3; % espacio de trabajo plano=3 espacio=6
%% Etapa 2 a Cinematica del Mecanismo de 5 barras
SMz = [0;0;1]; %Movimiento en z
S0a = [0;0;0]; %Vector de posición a
S0b = [bx;by;0]; %Vector de posición b
S0c = [cx;cy;0]; %Vector de posición c
S0e = [ex;ey;0]; %Vector de posición e
S0d = [dx;0;0]; %Vector de posición d
%% Etapa 3 a,b Cinematica del Mecanismo de 5 barras
f=1; %Movimiento unitario de los acoples
F=e*f; %Numero de movilidades
l=e-n+1; %Cantidad de circuitos
BM = [-1 -1 -1 -1 1]; %Matriz de circuitos del mecanismo de 5 barras
%% Etapa 4 a
sMa = [SMz*wa; cross(S0a,SMz*wa)]; %Helicoide de wa
sMb = [SMz*wb; cross(S0b,SMz*wb)]; %Helicoide de wb
sMc = [SMz*wc; cross(S0c,SMz*wc)]; %Helicoide de wc
sMe = [SMz*we; cross(S0e,SMz*we)]; %Helicoide de we
sMd = [SMz*wd; cross(S0d,SMz*wd)]; %Helicoide de wd
sMaa = [sMa(3);sMa(4);sMa(5)]; %Heligiros permitidos en a
sMbb = [sMb(3);sMb(4);sMb(5)]; %Heligiros permitidos en b
sMcc = [sMc(3);sMc(4);sMc(5)]; %Heligiros permitidos en c
sMee = [sMe(3);sMe(4);sMe(5)]; %Heligiros permitidos en e
sMdd = [sMd(3);sMd(4);sMd(5)]; %Heligiros permitidos en d
MD = [sMaa sMbb sMcc sMee sMdd]; %Matriz de movimientos
MDU = [sMaa/wa sMbb/wb sMcc/wc sMee/we sMdd/wd]; %Matriz de movimientos
unitarios
VT = [wa;wb;wc;we;wd]; %Vector de magnitudes
%% Etapa 5 a, b
MN = MDU*diag(BM); %Movimientos unitarios en red
MNEsc = rref(MN); %Matriz escalonada
%% Etapa 6 a b
m=3; %El numero de filas en diagonal
CN = (lambda*l)-m; %Grado de restriccion
MNEsc*VT;
%% Etapa 7 a, b
FN = F-m; %Grados de libertad
MNS = MNEsc(1:3,2:4); %Matriz de movimientos variables desconocidas
MNP = [MNEsc(:,1),MNEsc(:,5)]; %Matriz de moviientos variables conocidas
VTS = VT(2:4,1); %Vector de magnitudes desconocidas
```

```

VTP = [VT(1,1);VT(5,1)]; %Vector de magnitudes conocidas
%% Etapa 8 a
VTSR = -inv(MNS)*MNP*VTP; %Vector valores de las magnitudes desconocidas
%% Etapa 9 a
wbb = VTSR(1,1); %Velocidad angular en b
wcc = VTSR(2,1); %Velocidad angular en c
wee = VTSR(3,1); %Velocidad angular en e
sMaaa = MDU(1:3,1)*wa; % Helicide en a
sMbbb = MDU(1:3,2)*wbb; % Helicide en b
sMccc = MDU(1:3,3)*wcc; % Helicide en c
sMeee = MDU(1:3,4)*wee; % Helicide en e
sMddd = MDU(1:3,5)*wd; % Helicide en d
%% Etapa 2 a Estática del mecanismo de 5 barras
cp = 2; %Restricciones unitarias de esfuerzo (Rx,Ry)
ea = 3; %Acoples activos con 2 Tin y 1 Tout
%% Etapa 3 a, b
ca = 1; %Restriccion unitaria
SAX = [1;0;0]; %Helifuerzas dirección unitaria
SAY = [0;1;0]; %Helifuerzas dirección unitaria
SAZ = [0;0;1]; %Helifuerzas dirección unitaria
%Las posiciones S0 son iguales a los heligiros
C = (cp*e)+(ca*ea); %Grado bruto de restricciones
K = n-1; %Numero de cortes
QA = [1 1 0 0 0 0 0 0 1 1 1 1 0;...
      0 0 1 1 0 0 0 0 1 1 0 1 0;...
      0 0 0 0 1 1 0 0 1 1 0 1 1;...
      0 0 0 0 0 0 1 1 1 1 0 1 0]; %Matrix de cortes
%% Etapa 4 a, b
sAaRx = [cross(S0a,SAX*Rx);SAX*Rx]; %Helicoide de helifuerzas
sAaRy = [cross(S0a,SAY*Ray);SAY*Ray]; %Helicoide de helifuerzas
sAbRx = [cross(S0b,SAX*Rbx);SAX*Rbx]; %Helicoide de helifuerzas
sAbRy = [cross(S0b,SAY*Rby);SAY*Rby]; %Helicoide de helifuerzas
sAcRx = [cross(S0c,SAX*Rcx);SAX*Rcx]; %Helicoide de helifuerzas
sAcRy = [cross(S0c,SAY*Rcy);SAY*Rcy]; %Helicoide de helifuerzas
sAeRx = [cross(S0e,SAX*Rx);SAX*Rx]; %Helicoide de helifuerzas
sAeRy = [cross(S0e,SAY*Rey);SAY*Rey]; %Helicoide de helifuerzas
sAdRx = [cross(S0d,SAX*Rdx);SAX*Rdx]; %Helicoide de helifuerzas
sAdRy = [cross(S0d,SAY*Rdy);SAY*Rdy]; %Helicoide de helifuerzas
sAaTin = [SAZ*Ta;zeros(3,1)]; %Helicoide de Torque a
sAdTin = [SAZ*Td;zeros(3,1)]; %Helicoide de Torque d
sAcTout = [SAZ*Tc;zeros(3,1)]; %Helicoide de Torque c
sAaaRx = [sAaRx(3:5,1)]; %Helifuerzas son restrictas
sAaaRy = [sAaRy(3:5,1)]; %Helifuerzas son restrictas
sAbbRx = [sAbRx(3:5,1)]; %Helifuerzas son restrictas
sAbbRy = [sAbRy(3:5,1)]; %Helifuerzas son restrictas
sAccRx = [sAcRx(3:5,1)]; %Helifuerzas son restrictas
sAccRy = [sAcRy(3:5,1)]; %Helifuerzas son restrictas
sAeeRx = [sAeRx(3:5,1)]; %Helifuerzas son restrictas
sAeeRy = [sAeRy(3:5,1)]; %Helifuerzas son restrictas
sAddRx = [sAdRx(3:5,1)]; %Helifuerzas son restrictas
sAddRy = [sAdRy(3:5,1)]; %Helifuerzas son restrictas
sAaaTin = [sAaTin(3:5,1)]; %Helicoide de Torque a
sAddTin = [sAdTin(3:5,1)]; %Helicoide de Torque d
sAccTout = [sAcTout(3:5,1)]; %Helicoide de Torque c
AD = [sAaaRx sAaaRy sAbbRx sAbbRy sAccRx sAccRy sAeeRx sAeeRy sAddRx sAddRy
sAaaTin sAddTin sAccTout]; %Matriz de acciones

```

```

ADU = [sAaaRx/Rax sAaaRy/Ray sAbbRx/Rbx sAbbRy/Rby sAccRx/Rcx sAccRy/Rcy
sAeeRx/Rex sAeeRy/Rey sAddRx/Rdx sAddRy/Rdy sAaaTin/Ta sAddTin/Td
sAccTout/Tc]; %Matriz de acciones unitarias
VP = [Rax;Ray;Rbx;Rby;Rcx;Rcy;Rex;Rey;Rdx;Rdy;Ta;Td;Tc]; %Vector de acciones
%% Etapa 5 a, b
AN =
[ADU*diag(QA(1,1:13));ADU*diag(QA(2,1:13));ADU*diag(QA(3,1:13));ADU*diag(QA(4
,1:13))]; %Matriz de acciones unitarias en red
ANESC = rref(AN); %Matriz de acciones unitarias escalonada
%% Etapa 6 a, b
a=12;
FNN = (lambda*K)-a; %Libertad extra en el sistema
ANESC*VP;
%% Etapa 7 a, b
CNN = C-a; %Variabloes a despejar para hallar las demás
ANS = ANESC(:,1:12); %Variables a hallar
ANP = ANESC(:,13); %Variables conocidas
VPS = VP(1:12,1); %Variables a hallar
VPP = VP(13,1); %Variable conocida
%% Etapa 8 a
VPSR = -inv(ANS)*ANP*VPP; %Solucion
%% Etapa 9 a
Taa = VPSR(11) %Torque de entrada en a
Tdd = VPSR(12) %Torque de entrada en d

```

Anexo 20. Método de Davies mecanismo con pierna Matlab

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%UNIVERSIDAD DE PAMPLONA
%FACULTAD DE INGENIERIAS Y ARQUITECTURA
%MAESTRIA EN CONTROLES INDUSTRIALES
%DIRECTOR: CESAR AUGUSTO PEÑA CORTES
%ESTUDIANTE: JORGE ENRIQUE ARAQUE ISIDRO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, close all, clc
syms ax ay bx by cx cy ex ey dx dy fx fy gx gy hx hy wa wb wc we wd wf wg wh
Rax Ray Rbx Rby Rcx Rcy Rex Rey Rdx Rdy Rfx Rfy Rgx Rgy Rhx Rhy Ta Td Tf Tg
Th
%% Etapa 1 a,b y c
n=7; %numero de barras
e=8; %numero de acoples
lambda=3; % espacio de trabajo plano=3 espacio=6
%% Etapa 2 a Cinematica del Mecanismo de 5 barras
SMz = [0;0;1]; %Movimiento en z
S0a = [0;0;0]; %Vector de posición a
S0b = [bx;by;0]; %Vector de posición b
S0c = [cx;cy;0]; %Vector de posición c
S0e = [ex;ey;0]; %Vector de posición e
S0d = [dx;0;0]; %Vector de posición d
S0f = [fx;fy;0]; %Vector de posición f
S0g = [gx;gy;0]; %Vector de posición g
S0h = [hx;hy;0]; %Vector de posición h
%% Etapa 3 a,b Cinematica del Mecanismo de 5 barras
f=1; %Movimiento unitario de los acoples
F=e*f; %Numero de movilidades
l=e-n+1; %Cantidad de circuitos
BM = [-1 -1 -1 -1 1 0 0 0; -1 -1 0 0 0 -1 -1 1]; %Matriz de circuitos del
mecanismo de 5 barras
%% Etapa 4 a
sMa = [SMz*wa; cross(S0a,SMz*wa)]; %Helicoide de wa
sMb = [SMz*wb; cross(S0b,SMz*wb)]; %Helicoide de wb
sMc = [SMz*wc; cross(S0c,SMz*wc)]; %Helicoide de wc
sMe = [SMz*we; cross(S0e,SMz*we)]; %Helicoide de we
sMd = [SMz*wd; cross(S0d,SMz*wd)]; %Helicoide de wd
sMf = [SMz*wf; cross(S0f,SMz*wf)]; %Helicoide de wf
sMg = [SMz*wg; cross(S0g,SMz*wg)]; %Helicoide de wg
sMh = [SMz*wh; cross(S0h,SMz*wh)]; %Helicoide de wh
sMaa = [sMa(3);sMa(4);sMa(5)]; %Heligiros permitidos en a
sMbb = [sMb(3);sMb(4);sMb(5)]; %Heligiros permitidos en b
sMcc = [sMc(3);sMc(4);sMc(5)]; %Heligiros permitidos en c
sMee = [sMe(3);sMe(4);sMe(5)]; %Heligiros permitidos en e
sMdd = [sMd(3);sMd(4);sMd(5)]; %Heligiros permitidos en d
sMff = [sMf(3);sMf(4);sMf(5)]; %Heligiros permitidos en f
sMgg = [sMg(3);sMg(4);sMg(5)]; %Heligiros permitidos en g
sMhh = [sMh(3);sMh(4);sMh(5)]; %Heligiros permitidos en h
MD = [sMaa sMbb sMcc sMee sMdd sMff sMgg sMhh]; %Matriz de movimientos
MDU = [sMaa/wa sMbb/wb sMcc/wc sMee/we sMdd/wd sMff/wf sMgg/wg sMhh/wh];
%Matriz de movimientos unitarios
VT = [wa;wb;wc;we;wd;wf;wg;wh]; %Vector de magnitudes
%% Etapa 5 a, b

```

```

MN = [MDU*diag(BM(1,:));MDU*diag(BM(2,:))]; %Movimientos unitarios en red
MNESEC = rref(MN); %Matriz escalonada
%% Etapa 6 a b
m=6; %El numero de filas en diagonal
CN = (lambda*1)-m; %Grado de restriccion
MNESEC*VT;
%% Etapa 7 a, b
FN = F-m; %Grados de libertad
MNS = [MNESEC(:,2:4) MNESEC(:,6:8)]; %Matriz de movimientos variables
desconocidas
MNP = [MNESEC(:,1),MNESEC(:,5)]; %Matriz de moviientos variables conocidas
VTS = [VT(2:4,1);VT(6:8,1)]; %Vector de magnitudes desconocidas
VTP = [VT(1,1);VT(5,1)]; %Vector de magnitudes conocidas
%% Etapa 8 a
VTSR = -inv(MNS)*MNP*VTP; %Vector valores de las magnitudes desconocidas
%% Etapa 9 a
wbb = VTSR(1,1); %Velocidad angular en b
wcc = VTSR(2,1); %Velocidad angular en c
wee = VTSR(3,1); %Velocidad angular en e
wff = VTSR(4,1); %Velocidad angular en f
wee = VTSR(5,1); %Velocidad angular en g
wff = VTSR(6,1); %Velocidad angular en h
sMaaa = MDU(1:3,1)*wa; % Helicide en a
sMbbb = MDU(1:3,2)*wbb; % Helicide en b
sMccc = MDU(1:3,3)*wcc; % Helicide en c
sMeee = MDU(1:3,4)*wee; % Helicide en e
sMddd = MDU(1:3,5)*wd; % Helicide en d
sMfff = MDU(1:3,6)*wf; % Helicide en f
sMggg = MDU(1:3,7)*wg; % Helicide en g
sMhhh = MDU(1:3,8)*wh; % Helicide en h
%% Etapa 2 a Estática del mecanismo de 5 barras
cp = 2; %Restricciones unitarias de esfuerzo (Rx,Ry)
ea = 5; %Acoples activos con 2 Tin y 2 Tout
%% Etapa 3 a, b
ca = 1; %Restriccion unitaria
SAX = [1;0;0]; %Helifuerzas dirección unitaria
SAY = [0;1;0]; %Helifuerzas dirección unitaria
SAZ = [0;0;1]; %Helifuerzas dirección unitaria
%Las posiciones S0 son iguales a los heligiros
C = (cp*e)+(ca*ea); %Grado bruto de restricciones
K = n-1; %Numero de cortes
QA = [1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 1 1 1 0 0 1;...
      0 0 1 1 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 0 1;...
      0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 0 0 0;...
      0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 1 0 0 0;...
      0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1;...
      0 0 0 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 0 1]; %Matrix de cortes
%% Etapa 4 a, b
sAaRx = [cross(S0a,SAX*Rx);SAX*Rx]; %Helicoide de helifuerzas
sAaRy = [cross(S0a,SAY*Ray);SAY*Ray]; %Helicoide de helifuerzas
sAbRx = [cross(S0b,SAX*Rbx);SAX*Rbx]; %Helicoide de helifuerzas
sAbRy = [cross(S0b,SAY*Rby);SAY*Rby]; %Helicoide de helifuerzas
sAcRx = [cross(S0c,SAX*Rcx);SAX*Rcx]; %Helicoide de helifuerzas
sAcRy = [cross(S0c,SAY*Rcy);SAY*Rcy]; %Helicoide de helifuerzas
sAeRx = [cross(S0e,SAX*Rx);SAX*Rx]; %Helicoide de helifuerzas
sAeRy = [cross(S0e,SAY*Rey);SAY*Rey]; %Helicoide de helifuerzas
sAdRx = [cross(S0d,SAX*Rdx);SAX*Rdx]; %Helicoide de helifuerzas

```

```

sAdRy = [cross(S0d,SAY*Rdy);SAY*Rdy]; %Helicoide de helifuerzas
sAfRx = [cross(S0f,SAX*Rfx);SAX*Rfx]; %Helicoide de helifuerzas
sAfRy = [cross(S0f,SAY*Rfy);SAY*Rfy]; %Helicoide de helifuerzas
sAgRx = [cross(S0g,SAX*Rgx);SAX*Rgx]; %Helicoide de helifuerzas
sAgRy = [cross(S0g,SAY*Rgy);SAY*Rgy]; %Helicoide de helifuerzas
sAhRx = [cross(S0h,SAX*Rhx);SAX*Rhx]; %Helicoide de helifuerzas
sAhRy = [cross(S0h,SAY*Rhy);SAY*Rhy]; %Helicoide de helifuerzas
sAaTin = [SAZ*Ta;zeros(3,1)]; %Helicoide de Torque a
sAdTin = [SAZ*Td;zeros(3,1)]; %Helicoide de Torque d
sAfTout = [SAZ*Tf;zeros(3,1)]; %Helicoide de Torque f
sAgTout = [SAZ*Tg;zeros(3,1)]; %Helicoide de Torque g
sAhTout = [SAZ*Th;zeros(3,1)]; %Helicoide de Torque h
sAaaRx = [sAaRx(3:5,1)]; %Helifuerzas son restrictas
sAaaRy = [sAaRy(3:5,1)]; %Helifuerzas son restrictas
sAbbRx = [sAbRx(3:5,1)]; %Helifuerzas son restrictas
sAbbRy = [sAbRy(3:5,1)]; %Helifuerzas son restrictas
sAccRx = [sAcRx(3:5,1)]; %Helifuerzas son restrictas
sAccRy = [sAcRy(3:5,1)]; %Helifuerzas son restrictas
sAeeRx = [sAeRx(3:5,1)]; %Helifuerzas son restrictas
sAeeRy = [sAeRy(3:5,1)]; %Helifuerzas son restrictas
sAddRx = [sAdRx(3:5,1)]; %Helifuerzas son restrictas
sAddRy = [sAdRy(3:5,1)]; %Helifuerzas son restrictas
sAffRx = [sAfRx(3:5,1)]; %Helifuerzas son restrictas
sAffRy = [sAfRy(3:5,1)]; %Helifuerzas son restrictas
sAggRx = [sAgRx(3:5,1)]; %Helifuerzas son restrictas
sAggRy = [sAgRy(3:5,1)]; %Helifuerzas son restrictas
sAhhRx = [sAhRx(3:5,1)]; %Helifuerzas son restrictas
sAhhRy = [sAhRy(3:5,1)]; %Helifuerzas son restrictas
sAaaTin = [sAaTin(3:5,1)]; %Helicoide de Torque a
sAddTin = [sAdTin(3:5,1)]; %Helicoide de Torque d
sAffTout = [sAfTout(3:5,1)]; %Helicoide de Torque g
sAggTout = [sAgTout(3:5,1)]; %Helicoide de Torque g
sAhhTout = [sAhTout(3:5,1)]; %Helicoide de Torque h
AD = [sAaaRx sAaaRy sAbbRx sAbbRy sAccRx sAccRy sAeeRx sAeeRy sAddRx sAddRy
sAffRx sAffRy sAggRx sAggRy sAhhRx sAhhRy sAaaTin sAddTin sAffTout sAggTout
sAhhTout]; %Matriz de acciones
ADU = [sAaaRx/Rax sAaaRy/Ray sAbbRx/Rbx sAbbRy/Rby sAccRx/Rcx sAccRy/Rcy
sAeeRx/Rex sAeeRy/Rey sAddRx/Rdx sAddRy/Rdy sAffRx/Rfx sAffRy/Rfy sAggRx/Rgx
sAggRy/Rgy sAhhRx/Rhx sAhhRy/Rhy sAaaTin/Ta sAddTin/Td sAffTout/Tf
sAggTout/Tg sAhhTout/Th]; %Matriz de acciones unitarias
VP =
[Rax;Ray;Rbx;Rby;Rcx;Rcy;Rex;Rey;Rdx;Rdy;Rfx;Rfy;Rgx;Rgy;Rhx;Rhy;Ta;Td;Tf;Tg;
Th]; %Vector de acciones
%% Etapa 5 a, b
AN =
[ADU*diag(QA(1,:));ADU*diag(QA(2,:));ADU*diag(QA(3,:));ADU*diag(QA(4,:));ADU*
diag(QA(5,:));ADU*diag(QA(6,:))]; %Matriz de acciones unitarias en red
ANESC = rref(AN); %Matriz de acciones unitarias escalonada
%% Etapa 6 a, b
a=18;
FNN = (lambda*K)-a; %Libertad extra en el sistema
ANESC*VP;
%% Etapa 7 a, b
CNN = C-a; %Variabloes a despejar para hallar las demás
ANS = ANESC(:,1:18); %Variables a hallar
ANP = ANESC(:,19:21); %Variables conocidas
VPS = VP(1:18,1); %Variables a hallar

```

```
VPP = VP(19:21,1); %Variable conocida
%% Etapa 8 a
VPSR = -inv(ANS)*ANP*VPP; %Solucion
%% Etapa 9 a
Taa = VPSR(17) %Torque de entrada en a
Tdd = VPSR(18) %Torque de entrada en d
```


Anexo 21. Código programación Arduino Mega

```
#include <SPI.h>
#include <math.h>
#define GPIO_MISO 50
#define GPIO_MOSI 51
#define GPIO_SCK 52
#define GPIO_SS 53

//Angulos de la trayectoria Tray_Dir
float theta=0, alpha=0, beta=0;
//Posicion de la trayectoria Tray_Inv
float xt=0, yt=0;
//Posicion Trayectoria
float px[51], py[51], pxx[51], pyy[51], AM[51], pxxt[51], ppyt[51];
//Dimension del robot
float LB=10, LB1=17.3687, LB2=25.4656, LB3=17.3687, LB4=25.4656, LB5=28.0100,
ANG=40*(PI/180), Dx=39.6358, Dy=18.1680;
//Matrices de transformacion homogenea
//float MTHtrasx[4][4]={{ 1,0,0,Dx}, {0,1,0,0}, {0,0,1,0}, {0,0,0,1}};
//float MTHtrasy[4][4]={{ 1,0,0,0}, {0,1,0,Dy}, {0,0,1,0}, {0,0,0,1}};
//Angulos de los motores
float t1, t2, T1[51], T2[51];
uint16_t x=1;
uint16_t v[10];
int i=0, j=0;
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  pinMode(13,OUTPUT);
  pinMode(4,0);
  pinMode(5,0);
  pinMode(6,0);
  pinMode(7,0);
  pinMode(8,0);
  pinMode(9,0);
  digitalWrite(13,LOW);
  digitalWrite(4,0);
  digitalWrite(7,0);
  digitalWrite(8,0);
  digitalWrite(9,0);
  pinMode(GPIO_SS,INPUT);
  pinMode(GPIO_SCK,INPUT);
  pinMode(GPIO_MISO,OUTPUT);
  SPCR |= _BV(SPE);
  SPI.attachInterrupt();
}
```

```

}

ISR (SPI_STC_vect) // Interrupcion SPI
{
x = SPDR; //Obtenemos el bytebu
if(x!=0){
  if(x!=1){
    v[i]=x;
    i++;
  }
}
} //

void loop() {
// put your main code here, to run repeatedly:
if(x>=3){
  digitalWrite(13,HIGH);
  j=2;
}
else{
  digitalWrite(13,LOW);
  j=0;
}
//Serial.println(x);
if(i>=10 && j==2){
  for(int j=0;j<=9;j++){
    Serial.println(v[j]);
  }
  trayectoria(v[0],v[1]);
  Serial.println("trayectoria");
  for(int w=0;w<=50;w++){
    Serial.println("x");
    Serial.println(pxx[w]);
    Serial.println("y");
    Serial.println(pyy[w]);
    transformar_trayectoria(pxx[w],pyy[w],w);
  }
  Serial.println("Angulos de motores");
  for(int w=0;w<=50;w++){
    cin_inv(LB,LB1,LB2,LB3,LB4,LB5,ANG,pxxt[w],pyyt[w]);
    T1[w] = t1*(180/PI);
    T2[w] = t2*(180/PI);
    Serial.println("T1");
    Serial.println(t1);//T1[w]);
    Serial.println("T2");
    Serial.println(t2);//T2[w]);
  }
}

```

```

    }
    Serial.println("Arco de movilidad");
    for(int w=0;w<=50;w++){
        Serial.println(AM[w]);
    }
    Serial.println("Angulos del motor");
    for(int w=0;w<=50;w++){
        Serial.println("T1");
        Serial.println(T1[w]);
        Serial.println("T2");
        Serial.println(T2[w]);
    }
    i=0;
    j=0;
}

}

void trayectoria(float v0, float v1){
    int x=0, i=0;
    float y, px[50], py[50], L1, L2, CR, RT, p1, p2, p3, q1, q2;

    L1 = 36; //medida del paciente base de cadera-rodilla
    L2 = 36; //medida del paciente base de rodilla-tobillo
    CR = v0; //medida del paciente de cadera-rodilla
    RT = v1; //medida del paciente de rodilla-tobillo
    //Constantes de la curva que da la trayectoria
    p1 = -793.4;
    p2 = -3.047E5;
    p3 = 1.36E4;
    q1 = -1.981E4;
    q2 = 6.556E4;

    for(x=0;x>=-50;x--){
        y = ((p1*pow(x,2))+(p2*x)+p3)/(pow(x,2)+(q1*x)+q2);//Ecuación de la trayectoria
        px[i] = x;
        py[i] = y;
        Tray_Dir(L1,L2,x,y);
        Tray_Inv(CR,RT,theta,alpha,beta);
        pxx[i] = xt;
        pyy[i] = yt;
        AM[i] = 180-(beta*(180/PI));//Arco de movilidad
        i++;
    }
}

```

```

void Tray_Dir(int L1, int L2, int x, float y){
    float C, J, st, ct, ca, sa, cb, sb;
    x = x*(-1);
    if(x==0){
        theta = 0;
        alpha = 0;
        beta = PI;
    }
    else{
        C = L1+L2-x;
        J = sqrt(pow(C,2)+pow(y,2));
        st = y/J;
        ct = C/J;
        theta = atan2(st,ct);
        ca = (pow(L1,2)+pow(J,2)-pow(L2,2))/(2*L1*L2);
        sa = 1*sqrt(1-pow(ca,2));
        alpha = atan2(sa,ca);
        cb = (pow(L1,2)+pow(L2,2)-pow(J,2))/(2*L1*L2);
        sb = 1*sqrt(1-pow(cb,2));
        beta = atan2(sb,cb);
    }
}

```

```

void Tray_Inv(float L1, float L2, float theta, float alpha, float beta){
    float C, J;
    if(beta==PI){
        xt = 0;
        yt = 0;
    }
    else{
        J = sqrt(pow(L1,2)+pow(L2,2)-(2*L1*L2*cos(beta)));
        yt = J*sin(theta);
        C = J*cos(theta);
        xt = L1+L2-C;
        xt = xt*(-1);
    }
}

```

```

void transformar_trayectoria(float px, float py, int w){
    pxxt[w] = px+Dx;
    ppyt[w] = py+Dy;
}

```

```

void cin_inv(float LB, float LB1, float LB2, float LB3, float LB4, float LB5, float ANG, float
Fx, float Fy){

```

```

float CODO1, CODO2, Ax, Ay, Cx, Cy, a, i, t11, t22, v, cosalpha, senalpha, alphain, betain,
x15, y15, x12, y12, ac, lamda, yota, Px, Py, b, cosphi, sinphi, phi, cospsi, sinpsi, psi, betain1;
CODO1 = 1;
CODO2 = 1;
Ax = 0;
Ay = 0;
Cx = LB;
Cy = 0;
i = 1;
a = sqrt(pow((Ax-Fx),2)+pow((Ay-Fy),2));
if(a>LB1+LB5){
    t1=0;
    t2=0;
    v=0;
    return;
}
cosalpha = (pow(a,2)-pow(LB1,2)-pow(LB5,2))/(2*LB1*LB5);
senalpha = CODO1*sqrt(1-pow(cosalpha,2));
if(senalpha=='nan'){
    t1 = 0;
    t2 = 0;
    v = 0;
    return;
}
alphain=atan2(senalpha,cosalpha);
x15 = LB5*cos(alphain);
y15 = LB5*sin(alphain);
betain = atan2(y15,x15+LB1);
t1 = betain+atan2(Fy,Fx);
x12 = LB2*cos(alphain+ANG);
y12 = LB2*sin(alphain+ANG);
ac = sqrt(pow((LB2*cos(alphain+ANG)+LB1),2)+pow((LB2*sin(alphain+ANG)),2));
lamda = atan2(LB2*sin(alphain+ANG),LB2*cos(alphain+ANG)+LB1);
yota = t1-lamda;
Px = ac*cos(yota);
Py = ac*sin(yota);
b = sqrt(pow((Cx-Px),2)+pow((Cy-Py),2));
if(b>LB3+LB4){
    CODO1=CODO1*(-1);
    cosalpha = (pow(a,2)-pow(LB1,2)-pow(LB5,2))/(2*LB1*LB5);
    senalpha = CODO1*sqrt(1-pow(cosalpha,2));
    alphain=atan2(senalpha,cosalpha);
    x15 = LB5*cos(alphain);
    y15 = LB5*sin(alphain);
    betain = atan2(y15,x15+LB1);
    t1 = betain+atan2(Fy,Fx);
}

```

```

x12 = LB2*cos(alphain+ANG);
yl2 = LB2*sin(alphain+ANG);
ac = sqrt(pow((LB2*cos(alphain+ANG)+LB1),2)+pow((LB2*sin(alphain+ANG)),2));
lamda = atan2(LB2*sin(alphain+ANG),LB2*cos(alphain+ANG)+LB1);
yota = t1-lamda;
Px = ac*cos(yota);
Py = ac*sin(yota);
b = sqrt(pow((Cx-Px),2)+pow((Cy-Py),2));
}
if(b>LB3+LB4){
    t1 = 0;
    t2 = 0;
    v = 0;
    return;
}
cosphi = (pow(LB3,2)+pow(LB4,2)-pow(b,2))/(2*LB3*LB4);
sinphi = CODO2*sqrt(1-pow(cosphi,2));
if(sinphi=='nan'){
    CODO2 = CODO2*(-1);
    cosalpha = (pow(a,2)-pow(LB1,2)-pow(LB5,2))/(2*LB1*LB5);
    senalpha = CODO1*sqrt(1-pow(cosalpha,2));
if(senalpha=='nan'){
    t1 = 0;
    t2 = 0;
    v = 0;
    return;
}
}
alphain=atan2(senalpha,cosalpha);
x15 = LB5*cos(alphain);
yl5 = LB5*sin(alphain);
betain = atan2(yl5,x15+LB1);
t1 = betain+atan2(Fy,Fx);
x12 = LB2*cos(alphain+ANG);
yl2 = LB2*sin(alphain+ANG);
ac = sqrt(pow((LB2*cos(alphain+ANG)+LB1),2)+pow((LB2*sin(alphain+ANG)),2));
lamda = atan2(LB2*sin(alphain+ANG),LB2*cos(alphain+ANG)+LB1);
yota = t1-lamda;
Px = ac*cos(yota);
Py = ac*sin(yota);
b = sqrt(pow((Cx-Px),2)+pow((Cy-Py),2));
if(b>LB3+LB4){
    CODO1=CODO1*(-1);
    cosalpha = (pow(a,2)-pow(LB1,2)-pow(LB5,2))/(2*LB1*LB5);
    senalpha = CODO1*sqrt(1-pow(cosalpha,2));

alphain=atan2(senalpha,cosalpha);

```

```

x15 = LB5*cos(alphain);
yl5 = LB5*sin(alphain);
betain = atan2(yl5,x15+LB1);
t1 = betain+atan2(Fy,Fx);
x12 = LB2*cos(alphain+ANG);
yl2 = LB2*sin(alphain+ANG);
ac = sqrt(pow((LB2*cos(alphain+ANG)+LB1),2)+pow((LB2*sin(alphain+ANG)),2));
lamda = atan2(LB2*sin(alphain+ANG),LB2*cos(alphain+ANG)+LB1);
yota = t1-lamda;
Px = ac*cos(yota);
Py = ac*sin(yota);
b = sqrt(pow((Cx-Px),2)+pow((Cy-Py),2));
}
if(b>LB3+LB4){
    t1 = 0;
    t2 = 0;
    v = 0;
    return;
}
cosphi = (pow(LB3,2)+pow(LB4,2)-pow(b,2))/(2*LB3*LB4);
sinphi = CODO2*sqrt(1-pow(cosphi,2));
if(sinphi=='nan'){
    t1 = 0;
    t2 = 0;
    v = 0;
}
}
phi = atan2(sinphi,cosphi);
cospsi = (pow(b,2)-pow(LB3,2)-pow(LB4,2))/(2*LB3*LB4);
sinpsi = sqrt(1-pow(cospsi,2));
if(sinpsi=='nan'){
    t1 = 0;
    t2 = 0;
    v = 0;
}
psi = atan2(sinpsi,cospsi);
betain1 = atan2(LB4*sin(psi),LB4*cos(psi)+LB3);
t2 = atan2((Py-Cy),(Px-Cx))-betain1;
}

```

Anexo 22. Código programación Arduino ESP32

```
#include <SPI.h>
#include <WiFi.h>

uint16_t x1=10;
uint16_t x2=20;
//-----Servidor Web en puerto 80-----

WiFiServer server(80);

//-----Credenciales de WiFi-----

const char* ssid = "TVNwifiAraqueB";
const char* password = "TVN60252684MIYG";

//-----VARIABLES GLOBALES-----
int contconexion = 0, L1, L2, Ang, C;

String header; // Variable para guardar el HTTP request

String estadoSalida = "off";

String valueString = String(5);
int pos1 = 0;
int pos2 = 0;
String valueString1 = String(5);
int pos3 = 0;
int pos4 = 0;
String valueString3 = String(5);
int pos5 = 0;
int pos6 = 0;
String valueString4 = String(5);
int pos7 = 0;
int pos8 = 0;

const int salida = 2;

//-----CODIGO HTML-----
String pagina = "<!DOCTYPE html>"
"<html>"
"<head>"
"<meta name=\"viewport\" content=\"width=device-width, initial-scale=1\">"
"<link rel=\"icon\" href=\"data:,\">>"
"<style>body { text-align: center; font-family: \"Trebuchet MS\", Arial; margin-left:auto; margin-right:auto;}"
```



```

.slider { width: 300px; }"
"html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}"
".button { background-color: #4CAF50; border: none; color: white; padding: 16px 40px;"
"text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}"
".button2 {background-color: #555555;}"
".ho {margin:auto; width:500px; padding:0; list-style:none; display:inline-block;}</style>"
"<script src=\"https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js\"></script>"
"<title>Robot de Rehabilitacion de Rodilla</title>"
"</head><body><h1>REHABOT</h1>"
"<p><a href='/on'><button style='height:50px;width:100px'>ON</button></a></p>"
"<p><a href='/off'><button style='height:50px;width:100px'>OFF</button></a></p>"
"<p>Cadera-Rodilla: <span id=\"Pos111\"></span></p>"
"<input type=\"range\" min=\"0\" max=\"100\" class=\"slider\" id=\"Slider1\"
onchange=\"servo(this.value)\" value=\"\"+valueString+\"\"/>"
"<script>var slider = document.getElementById(\"Slider1\");"
"var servoP = document.getElementById(\"Pos111\"); servoP.innerHTML = slider.value;"
"slider.oninput = function() { slider.value = this.value; servoP.innerHTML = this.value; }"
"$.ajaxSetup({timeout:1000}); function servo(pos11) { "
"$.get(\"/?value=\" + pos11 + \"&\"); {Connection: close;}</script>"
"<p>Rodilla-Tobillo: <span id=\"Pos222\"></span></p>"
"<input type=\"range\" min=\"0\" max=\"100\" class=\"slider\" id=\"Slider2\"
onchange=\"servo2(this.value)\" value=\"\"+valueString1+\"\"/>"
"<script>var slider2 = document.getElementById(\"Slider2\");"
"var servoP1 = document.getElementById(\"Pos222\"); servoP1.innerHTML = slider2.value;"
"slider2.oninput = function() { slider2.value = this.value; servoP1.innerHTML = this.value; }"
"$.ajaxSetup({timeout:1000}); function servo2(pos22) { "
"$.get(\"/?value1=\" + pos22 + \"&\"); {Connection: close;}</script>"
"<p>Angulo de Arco de Movilidad: <span id=\"Pos333\"></span></p>"
"<input type=\"range\" min=\"0\" max=\"180\" class=\"slider\" id=\"Slider3\"
onchange=\"servo3(this.value)\" value=\"\"+valueString3+\"\"/>"
"<script>var slider3 = document.getElementById(\"Slider3\");"
"var servoP3 = document.getElementById(\"Pos333\"); servoP3.innerHTML = slider3.value;"
"slider3.oninput = function() { slider3.value = this.value; servoP3.innerHTML = this.value; }"
"$.ajaxSetup({timeout:1000}); function servo3(pos33) { "
"$.get(\"/?value3=\" + pos33 + \"&\"); {Connection: close;}</script>"
"<p>Ciclos de Movimientos: <span id=\"Pos444\"></span></p>"
"<input type=\"range\" min=\"0\" max=\"20\" class=\"slider\" id=\"Slider4\"
onchange=\"servo4(this.value)\" value=\"\"+valueString4+\"\"/>"
"<script>var slider4 = document.getElementById(\"Slider4\");"
"var servoP4 = document.getElementById(\"Pos444\"); servoP4.innerHTML = slider4.value;"
"slider4.oninput = function() { slider4.value = this.value; servoP4.innerHTML = this.value; }"
"$.ajaxSetup({timeout:1000}); function servo4(pos44) { "
"$.get(\"/?value4=\" + pos44 + \"&\"); {Connection: close;}</script>"
"</body>"
"</html>";

```

```

//-----SETUP-----
void setup() {
  Serial.begin(115200);
  Serial.println("");

  //pinMode(SpiCsPin, OUTPUT);
  //digitalWrite(SpiCsPin, HIGH);
  SPI.begin();
  SPI.setClockDivider(SPI_CLOCK_DIV8);
  pinMode(5,OUTPUT);
  pinMode(18,OUTPUT);
  digitalWrite(18,LOW);
  digitalWrite(5,HIGH);
  //SPI.beginTransaction(SPISettings(SpiSpeed, MSBFIRST, SPI_MODE0));

  pinMode(salida, OUTPUT);
  digitalWrite(salida, LOW);

  // Conexión WIFI
  WiFi.begin(ssid, password);
  //Cuenta hasta 50 si no se puede conectar lo cancela
  while (WiFi.status() != WL_CONNECTED and contconexion <50) {
    ++contconexion;
    delay(500);
    Serial.print(".");
  }
  if (contconexion <50) {
    //para usar con ip fija
    //IPAddress ip(192,168,1,180);
    //IPAddress gateway(192,168,1,1);
    //IPAddress subnet(255,255,255,0);
    //WiFi.config(ip, gateway, subnet);

    Serial.println("");
    Serial.println("WiFi conectado");
    Serial.println(WiFi.localIP());
    server.begin(); // iniciamos el servidor
  }
  else {
    Serial.println("");
    Serial.println("Error de conexion");
  }
}

//-----LOOP-----

```

```

void loop(){
  WiFiClient client = server.available(); // Escucha a los clientes entrantes

  if (client) { // Si se conecta un nuevo cliente
    Serial.println("New Client."); //
    String currentLine = ""; //
    while (client.connected()) { // loop mientras el cliente está conectado
      if (client.available()) { // si hay bytes para leer desde el cliente
        char c = client.read(); // lee un byte
        Serial.write(c); // imprime ese byte en el monitor serial
        header += c;
        if (c == '\n') { // si el byte es un caracter de salto de linea
          // si la nueva linea está en blanco significa que es el fin del
          // HTTP request del cliente, entonces respondemos:
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println("Connection: close");
            client.println();

            //client.println(pagina);
            //client.println();

            // enciende y apaga el GPIO
            if (header.indexOf("GET /on") >= 0) {
              Serial.println("GPIO on");
              estadoSalida = "on";
              digitalWrite(salida, HIGH);
              digitalWrite(5, LOW);
              delay(50);
              SPI.transfer16(L1);
              delay(50);
              SPI.transfer16(L2);
              delay(50);
              SPI.transfer16(Ang);
              delay(50);
              SPI.transfer16(C);
              delay(50);
              SPI.transfer16(50);
              delay(50);
              SPI.transfer16(60);
              delay(50);
              SPI.transfer16(70);
              delay(50);
            }
          }
        }
      }
    }
  }
}

```

```

SPI.transfer16(80);
delay(50);
SPI.transfer16(90);
delay(50);
SPI.transfer16(100);
delay(50);
digitalWrite(5, HIGH);
} else if (header.indexOf("GET /off") >= 0) {
Serial.println("GPIO off");
estadoSalida = "off";
digitalWrite(salida, LOW);
digitalWrite(5, LOW);
delay(50);
SPI.transfer16(1);
delay(50);
digitalWrite(5, HIGH);
}

// Muestra la página web
//client.println(pagina);

if(header.indexOf("GET /?value=")>=0) {
pos1 = header.indexOf('=');
pos2 = header.indexOf('&');
valueString = header.substring(pos1+1, pos2);
L1=String(valueString).toInt();
}

if(header.indexOf("GET /?value1=")>=0) {
pos3 = header.indexOf('=');
pos4 = header.indexOf('&');
valueString1 = header.substring(pos3+1, pos4);
L2=String(valueString1).toInt();
}

if(header.indexOf("GET /?value3=")>=0) {
pos5 = header.indexOf('=');
pos6 = header.indexOf('&');
valueString3 = header.substring(pos5+1, pos6);
Ang=String(valueString3).toInt();
}

if(header.indexOf("GET /?value4=")>=0) {
pos7 = header.indexOf('=');
pos8 = header.indexOf('&');
valueString4 = header.substring(pos7+1, pos8);
}

```

```

    C=String(valueString4).toInt();
  }

  client.println("<!DOCTYPE html><html>");
  client.println("<head><meta name=\"viewport\" content=\"width=device-width, initial-
scale=1\">");
  client.println("<link rel=\"icon\" href=\"data:,\">");
  // CSS to style the on/off buttons
  // Feel free to change the background-color and font-size attributes to fit your preferences
  client.println("<style>body { text-align: center; font-family: \"Trebuchet MS\", Arial;
margin-left:auto; margin-right:auto;}");
  client.println(".slider { width: 300px; }");
  client.println("html { font-family: Helvetica; display: inline-block; margin: 0px auto;
text-align: center;}");
  client.println(".button { background-color: #4CAF50; border: none; color: white;
padding: 16px 40px;");
  client.println("text-decoration: none; font-size: 30px; margin: 2px; cursor: pointer;}");
  client.println(".button2 {background-color: #555555;}");
  client.println(".ho {margin:auto; width:500px; padding:0; list-style:none; display:inline-
block;}</style>");
  client.println("<script
src=\"https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js\"></script>");
  client.println("<title>Robot de Rehabilitacion de Rodilla</title>");
  client.println("</head><body><h1>REHABOT</h1>");
  client.println("<p><a href='/on'><button
style='height:50px;width:100px'>ON</button></a></p>");
  client.println("<p><a href='/off'><button
style='height:50px;width:100px'>OFF</button></a></p>");
  client.println("<p>Cadera-Rodilla: <span id=\"Pos111\"></span></p>");
  client.println("<input type=\"range\" min=\"0\" max=\"100\" class=\"slider\"
id=\"Slider1\" onchange=\"servo(this.value)\" value=\""+valueString+"\"/>");
  client.println("<script>var slider = document.getElementById(\"Slider1\");");
  client.println("var servoP = document.getElementById(\"Pos111\"); servoP.innerHTML
= slider.value;");
  client.println("slider.oninput = function() { slider.value = this.value; servoP.innerHTML
= this.value; }");
  client.println("$.ajaxSetup({timeout:1000}); function servo(pos11) { ");
  client.println("$.get(\"/?value=\" + pos11 + \"&\"); {Connection: close};}</script>");
  client.println("<p>Rodilla-Tobillo: <span id=\"Pos222\"></span></p>");
  client.println("<input type=\"range\" min=\"0\" max=\"100\" class=\"slider\"
id=\"Slider2\" onchange=\"servo2(this.value)\" value=\""+valueString1+"\"/>");
  client.println("<script>var slider2 = document.getElementById(\"Slider2\");");
  client.println("var servoP1 = document.getElementById(\"Pos222\");
servoP1.innerHTML = slider2.value;");
  client.println("slider2.oninput = function() { slider2.value = this.value;
servoP1.innerHTML = this.value; }");

```

```

client.println("$.ajaxSetup({timeout:1000}); function servo2(pos22) { ");
client.println("$.get(\"/?value1=\" + pos22 + \"&\"); {Connection: close;}</script>");
client.println("<p>Angulo de Arco de Movilidad: <span id=\"Pos333\"></span></p>");
client.println("<input type=\"range\" min=\"0\" max=\"180\" class=\"slider\"
id=\"Slider3\" onchange=\"servo3(this.value)\" value=\""+valueString3+"\"/>");
client.println("<script>var slider3 = document.getElementById(\"Slider3\");");
client.println("var servoP3 = document.getElementById(\"Pos333\");
servoP3.innerHTML = slider3.value;");
client.println("slider3.oninput = function() { slider3.value = this.value;
servoP3.innerHTML = this.value; }");
client.println("$.ajaxSetup({timeout:1000}); function servo3(pos33) { ");
client.println("$.get(\"/?value3=\" + pos33 + \"&\"); {Connection: close;}</script>");
client.println("<p>Ciclos de Movimientos: <span id=\"Pos444\"></span></p>");
client.println("<input type=\"range\" min=\"0\" max=\"20\" class=\"slider\" id=\"Slider4\"
onchange=\"servo4(this.value)\" value=\""+valueString4+"\"/>");
client.println("<script>var slider4 = document.getElementById(\"Slider4\");");
client.println("var servoP4 = document.getElementById(\"Pos444\");
servoP4.innerHTML = slider4.value;");
client.println("slider4.oninput = function() { slider4.value = this.value;
servoP4.innerHTML = this.value; }");
client.println("$.ajaxSetup({timeout:1000}); function servo4(pos44) { ");
client.println("$.get(\"/?value4=\" + pos44 + \"&\"); {Connection: close;}</script>");
client.println("</body>");
client.println("</html>");

// la respuesta HTTP termina con una linea en blanco
client.println();
break;
} else { // si tenemos una nueva linea limpiamos currentLine
currentLine = "";
}
} else if (c != '\r') { // si C es distinto al caracter de retorno de carro
currentLine += c; // lo agrega al final de currentLine
}
}
}
// Limpiamos la variable header
header = "";
// Cerramos la conexión
client.stop();
Serial.println("Client disconnected.");
Serial.println("");
}
}

```

Anexo 23 Código programación prueba de encoder Arduino Mega

```
double i=0, j=0, x=0, y=0;
//volatile unsigned int temp, counter = 0, counter1 = 0; //This variable will increase or decrease
depending on the rotation of encoder
int temp, counter = 0, counter1 = 0; //This variable will increase or decrease depending on the
rotation of encoder
    double q=600, T=1, ang = 0, ang1 = 0, wang = 0, rpm = 0;
float
ar1[51]={45,45,46,48,51,55,59,63,67,71,74,78,81,84,88,91,94,97,100,103,106,108,111,114,117,
119,122,125,127,130,132,135,137,140,142,144,147,149,151,153,155,156,158,160,161,163,164,1
65,166,168,168};
float ar2[51]={-71,-59,-53,-50,-48,-48,-48,-48,-48,-48,-49,-49,-50,-50,-50,-50,-50,-50,-50,-
49,-49,-48,-47,-46,-45,-44,-42,-41,-39,-37,-34,-32,-30,-27,-24,-21,-18,-15,-12,-9,-6,-
3,1,4,7,10,13,16,19,23};
void setup() {
    Serial.begin (115200);

    pinMode(21, INPUT_PULLUP); // internal pullup input pin 21

    pinMode(20, INPUT_PULLUP); // internal pullup input pin 20

    pinMode(19, INPUT_PULLUP); // internal pullup input pin 19

    pinMode(18, INPUT_PULLUP); // internal pullup input pin 18
//Setting up interrupt
//A rising pulse from encodenren activated ai0(). AttachInterrupt 0 is DigitalPin nr 2 on moust
Arduino.
    attachInterrupt(2, ai0, RISING);

//B rising pulse from encodenren activated ai1(). AttachInterrupt 1 is DigitalPin nr 3 on moust
Arduino.
// attachInterrupt(3, ai1, RISING);

//C rising pulse from encodenren activated ai0(). AttachInterrupt 0 is DigitalPin nr 2 on moust
Arduino.
    attachInterrupt(4, ai2, RISING);

//D rising pulse from encodenren activated ai1(). AttachInterrupt 1 is DigitalPin nr 3 on moust
Arduino.
//attachInterrupt(5, ai3, RISING);
// put your setup code here, to run once:
for(int j=4;j<=9;j++){
    pinMode(j,0);
}
Serial.begin(115200);
```

```

digitalWrite(4,0);
digitalWrite(7,0);
digitalWrite(8,0);
digitalWrite(9,0);
delay(2000);
x=300;
Serial.println (ar1[1]-ar1[0]);
Serial.println (ar2[1]-ar2[0]);
/*for(int z=0;z<=50;z++){
  Serial.println (ar1[z]);
  Serial.println (ar2[z]);
}*/
}

void loop() {
  // put your main code here, to run repeatedly:
  ang = (360*counter)/q;
  ang1 = (360*counter1)/q;
  Serial.print("angulo:");
  Serial.println (ang);
  Serial.print("angulo1:");
  Serial.println (ang1);
  if(i<=5){
    if(ang<=(ar1[51]-ar1[0])){
      digitalWrite(7,1);
      digitalWrite(9,0);
    }
    else{
      digitalWrite(7,0);
      digitalWrite(9,0);
    }
    if(ang1<=(ar2[51]-ar2[0])){
      digitalWrite(4,0);
      digitalWrite(8,1);
    }
    else{
      digitalWrite(4,0);
      digitalWrite(8,0);
    }
    analogWrite(5,255);
    analogWrite(6,255);
  }
  else{
    digitalWrite(4,0);
    digitalWrite(7,0);
    digitalWrite(8,0);
  }
}

```



```

digitalWrite(9,0);
analogWrite(5,0);
analogWrite(6,0);
}
if(i>=500){
  i=0;
}
i++;
}

```

```

void ai0() {
// ai0 is activated if DigitalPin nr 2 is going from LOW to HIGH
// Check pin 3 to determine the direction
if(digitalRead(20)==LOW) {
counter++;//++ cable gris
}else{
counter--;//-- cable gris
}
}
}

```

```

/* void ai1() {
// ai0 is activated if DigitalPin nr 3 is going from LOW to HIGH
// Check with pin 2 to determine the direction
if(digitalRead(21)==LOW) {
counter++;//-- cable gris
}else{
counter--;//++ cable gris
}
}
}*/

```

```

void ai2() {
// ai0 is activated if DigitalPin nr 2 is going from LOW to HIGH
// Check pin 3 to determine the direction
if(digitalRead(18)==LOW) {
counter1++;//++ cable gris
}else{
counter1--;//-- cable gris
}
}
}

```

```

/*void ai3() {
// ai0 is activated if DigitalPin nr 3 is going from LOW to HIGH
// Check with pin 2 to determine the direction
if(digitalRead(19)==LOW) {
counter1++;//-- cable gris
}else{

```

```
counter1--;//++ cable gris
}
}*/
```

Anexo 22 Código programación prueba de velocidad motores Arduino Mega

```
double i=0, j=0;
void setup() {

// put your setup code here, to run once:
for(int j=4;j<=9;j++){
  pinMode(j,0);
}
Serial.begin(115200);
```

```

digitalWrite(4,0);
digitalWrite(7,0);
digitalWrite(8,0);
digitalWrite(9,0);
delay(2000);
}

void loop() {
// put your main code here, to run repeatedly:
if(i<=30){
digitalWrite(4,0);
digitalWrite(7,0);
digitalWrite(8,1);
digitalWrite(9,0);
analogWrite(5,255);
analogWrite(6,255);
}
else{
digitalWrite(4,0);
digitalWrite(7,0);
digitalWrite(8,0);
digitalWrite(9,0);
analogWrite(5,0);
analogWrite(6,0);
}
if(i>=1000){
i=0;
}
i++;
}

```

Anexo 23. Código programación prueba de encoders y motores Arduino Mega

```

volatile unsigned int temp, counter = 0, counter1 = 0, i = 0; //This variable will increase or
decrease depending on the rotation of encoder
double q=1200, T=1, ang = 0, wang = 0, rpm = 0;
void setup() {
Serial.begin (115200);

pinMode(2, INPUT_PULLUP); // internal pullup input pin 2

```

```

pinMode(3, INPUT_PULLUP); // internal pullup input pin 3
//Setting up interrupt
//A rising pulse from encodenren activated ai0(). AttachInterrupt 0 is DigitalPin nr 2 on moust
Arduino.
attachInterrupt(0, ai0, RISING);

//B rising pulse from encodenren activated ai1(). AttachInterrupt 1 is DigitalPin nr 3 on moust
Arduino.
attachInterrupt(1, ai1, RISING);
}

void loop() {
// Send the value of counter
/*if( counter != temp ){
Serial.println (counter);
temp = counter;
}
*/
if(counter != counter1){
i++;
}
ang = (2*3.14159*counter)/q;
wang = (2*3.14159*(counter-counter1))/(q*0.001);
counter1 = counter;
if(i==1000){
rpm = (counter/q)*60;
}
Serial.print("angulo:");
Serial.println (ang);
Serial.print("\tvelocidad angular:");
Serial.println (wang);
Serial.print("rpm:");
Serial.println (rpm);
delay(1);
}
void ai0() {
// ai0 is activated if DigitalPin nr 2 is going from LOW to HIGH
// Check pin 3 to determine the direction
if(digitalRead(3)==LOW) {
counter--;//++ cable gris
}else{
counter++;/-- cable gris
}
}
}

```

```
void ai1() {  
  // ai0 is activated if DigitalPin nr 3 is going from LOW to HIGH  
  // Check with pin 2 to determine the direction  
  if(digitalRead(2)==LOW) {  
    counter++;/-- cable gris  
  }else{  
    counter--;++ cable gris  
  }  
}
```

Anexo 24. Estapa 8a mecaniso con pierna

$$\begin{aligned} & (Tf*(bx*cx*gy - bx*cy*gx - bx*cx*hy + bx*cy*hx - bx*ex*gy + bx*ey*gx + bx*ex*hy - \\ & bx*ey*hx - cx*ey*gx + cy*ex*gx + cx*ey*hx - cy*ex*hx + cx*gx*hy - cx*gy*hx - ex*gx*hy + \\ & ex*gy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + \\ & gx*hy - gy*hx)) - (Tg*(bx*cx*fy - bx*cy*fx - bx*cx*hy + bx*cy*hx - bx*ex*fy + bx*ey*fx - \\ & cx*ey*fx + cy*ex*fx + bx*ex*hy - bx*ey*hx + cx*ey*hx - cy*ex*hx + cx*fx*hy - cx*fy*hx - \\ & ex*fx*hy + ex*fy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - \\ & fx*hy + fy*hx + gx*hy - gy*hx)) - (Th*(bx*cx*fy - bx*cy*fx - bx*cx*gy + bx*cy*gx - \\ & bx*ex*fy + bx*ey*fx + bx*ex*gy - bx*ey*gx - cx*ey*fx + cy*ex*fx + cx*ey*gx - cy*ex*gx + \\ & cx*fx*gy - cx*fy*gx - ex*fx*gy + ex*fy*gx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - \\ & cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)) \end{aligned}$$

$$\begin{aligned} & (Tf*(by*cx*gy - by*cy*gx - by*cx*hy + by*cy*hx - by*ex*gy + by*ey*gx + by*ex*hy - \\ & by*ey*hx - cx*ey*gy + cy*ex*gy + cx*ey*hy - cy*ex*hy + cy*gx*hy - cy*gy*hx - ey*gx*hy + \\ & ey*gy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + \\ & gx*hy - gy*hx)) - (Tg*(by*cx*fy - by*cy*fx - by*cx*hy + by*cy*hx - by*ex*fy + by*ey*fx - \\ & cx*ey*fy + cy*ex*fy + by*ex*hy - by*ey*hx + cx*ey*hy - cy*ex*hy + cy*fx*hy - cy*fy*hx - \\ & ey*fx*hy + ey*fy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - \\ & fx*hy + fy*hx + gx*hy - gy*hx)) - (Th*(by*cx*fy - by*cy*fx - by*cx*gy + by*cy*gx - \\ & by*ex*fy + by*ey*fx + by*ex*gy - by*ey*gx - cx*ey*fy + cy*ex*fy + cx*ey*gy - cy*ex*gy + \\ & cy*fx*gy - cy*fy*gx - ey*fx*gy + ey*fy*gx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - \\ & cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)) \end{aligned}$$

$$\begin{aligned} & (Tf*(bx*cx*gy - bx*cy*gx - bx*cx*hy + bx*cy*hx - bx*ex*gy + bx*ey*gx + bx*ex*hy - \\ & bx*ey*hx - cx*ey*gx + cy*ex*gx + cx*ey*hx - cy*ex*hx + cx*gx*hy - cx*gy*hx - ex*gx*hy + \\ & ex*gy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + \\ & gx*hy - gy*hx)) - (Tg*(bx*cx*fy - bx*cy*fx - bx*cx*hy + bx*cy*hx - bx*ex*fy + bx*ey*fx - \\ & cx*ey*fx + cy*ex*fx + bx*ex*hy - bx*ey*hx + cx*ey*hx - cy*ex*hx + cx*fx*hy - cx*fy*hx - \\ & ex*fx*hy + ex*fy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - \\ & fx*hy + fy*hx + gx*hy - gy*hx)) - (Th*(bx*cx*fy - bx*cy*fx - bx*cx*gy + bx*cy*gx - \\ & bx*ex*fy + bx*ey*fx + bx*ex*gy - bx*ey*gx - cx*ey*fx + cy*ex*fx + cx*ey*gx - cy*ex*gx + \\ & cx*fx*gy - cx*fy*gx - ex*fx*gy + ex*fy*gx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - \\ & cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)) \end{aligned}$$

$$\begin{aligned} & (Tf*(by*cx*gy - by*cy*gx - by*cx*hy + by*cy*hx - by*ex*gy + by*ey*gx + by*ex*hy - \\ & by*ey*hx - cx*ey*gy + cy*ex*gy + cx*ey*hy - cy*ex*hy + cy*gx*hy - cy*gy*hx - ey*gx*hy + \\ & ey*gy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + \\ & gx*hy - gy*hx)) - (Tg*(by*cx*fy - by*cy*fx - by*cx*hy + by*cy*hx - by*ex*fy + by*ey*fx - \\ & cx*ey*fy + cy*ex*fy + by*ex*hy - by*ey*hx + cx*ey*hy - cy*ex*hy + cy*fx*hy - cy*fy*hx - \\ & ey*fx*hy + ey*fy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - \\ & fx*hy + fy*hx + gx*hy - gy*hx)) - (Th*(by*cx*fy - by*cy*fx - by*cx*gy + by*cy*gx - \\ & by*ex*fy + by*ey*fx + by*ex*gy - by*ey*gx - cx*ey*fy + cy*ex*fy + cx*ey*gy - cy*ex*gy + \\ & cy*fx*gy - cy*fy*gx - ey*fx*gy + ey*fy*gx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - \\ & cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)) \end{aligned}$$

$$fx*hy + fy*hx + gx*hy - gy*hx) - (Tf*(bx*cx*gy - by*cx*gx - bx*cx*hy + by*cx*hx - bx*ex*gy + by*ex*gx + bx*ex*hy - by*ex*hx + cx*gx*hy - cx*gy*hx - ex*gx*hy + ex*gy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx))$$

$$(Th*(bx*cy*fy - by*cy*fx - bx*cy*gy + by*cy*gx - bx*ey*fy + by*ey*fx + bx*ey*gy - by*ey*gx + cy*fx*gy - cy*fy*gx - ey*fx*gy + ey*fy*gx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)) + (Tg*(bx*cy*fy - by*cy*fx - bx*cy*hy - bx*ey*fy + by*cy*hx + by*ey*fx + bx*ey*hy - by*ey*hx + cy*fx*hy - cy*fy*hx - ey*fx*hy + ey*fy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) - (Tf*(bx*cy*gy - by*cy*gx - bx*cy*hy + by*cy*hx - bx*ey*gy + by*ey*gx + bx*ey*hy - by*ey*hx + cy*gx*hy - cy*gy*hx - ey*gx*hy + ey*gy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx))$$

$$(Th*(fx - gx))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) + (Tg*(fx - hx))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) - (Tf*(gx - hx))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)$$

$$(Th*(fy - gy))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) + (Tg*(fy - hy))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) - (Tf*(gy - hy))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)$$

$$(Th*(fx - gx))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) + (Tg*(fx - hx))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) - (Tf*(gx - hx))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)$$

$$(Th*(fy - gy))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) + (Tg*(fy - hy))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) - (Tf*(gy - hy))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)$$

$$(Tf*(gx - hx))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) - (Tg*(fx - hx))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) - (Th*(fx - gx))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)$$

$$(Tf*(gy - hy))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) - (Tg*(fy - hy))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) - (Th*(fy - gy))/(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)$$

$$(Th*(bx*cx*ey*fy - bx*cy*ex*fy - by*cx*ey*fx + by*cy*ex*fx - bx*cx*ey*gy + bx*cy*ex*gy + by*cx*ey*gx - by*cy*ex*gx - bx*cy*fx*gy + bx*cy*fy*gx + by*cx*fx*gy - by*cx*fy*gx + bx*ey*fx*gy - bx*ey*fy*gx - by*ex*fx*gy + by*ex*fy*gx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)) + (Tg*(bx*cx*ey*fy - bx*cy*ex*fy - by*cx*ey*fx + by*cy*ex*fx - bx*cx*ey*hy + bx*cy*ex*hy + by*cx*ey*hx - by*cy*ex*hx - bx*cy*fx*hy + bx*cy*fy*hx + by*cx*fx*hy - by*cx*fy*hx + bx*ey*fx*hy - bx*ey*fy*hx - by*ex*fx*hy + by*ex*fy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx) - (Tf*(bx*cx*ey*gy - bx*cy*ex*gy -$$

$$\begin{aligned} &by*cx*ey*gx + by*cy*ex*gx - bx*cx*ey*hy + bx*cy*ex*hy + by*cx*ey*hx - by*cy*ex*hx - \\ &bx*cy*gx*hy + bx*cy*gy*hx + by*cx*gx*hy - by*cx*gy*hx + bx*ey*gx*hy - bx*ey*gy*hx - \\ &by*ex*gx*hy + by*ex*gy*hx)/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - \\ &fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)) \end{aligned}$$

$$\begin{aligned} &(Tf*(cy*dx + cx*ey - cy*ex - dx*ey)*(bx*gy - by*gx - bx*hy + by*hx + gx*hy - \\ &gy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + \\ &gx*hy - gy*hx)) - (Tg*(cy*dx + cx*ey - cy*ex - dx*ey)*(bx*fy - by*fx - bx*hy + by*hx + fx*hy \\ &- fy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + \\ &gx*hy - gy*hx)) - (Th*(cy*dx + cx*ey - cy*ex - dx*ey)*(bx*fy - by*fx - bx*gy + by*gx + fx*gy \\ &- fy*gx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + \\ &gx*hy - gy*hx)) \end{aligned}$$

Anexo 25. Etapa 9a mecanismo con pierna

T1 =

$$\begin{aligned} & (Th*(bx*cx*ey*fy - bx*cy*ex*fy - by*cx*ey*fx + by*cy*ex*fx - bx*cx*ey*gy + bx*cy*ex*gy \\ & + by*cx*ey*gx - by*cy*ex*gx - bx*cy*fx*gy + bx*cy*fy*gx + by*cx*fx*gy - by*cx*fy*gx + \\ & bx*ey*fx*gy - bx*ey*fy*gx - by*ex*fx*gy + by*ex*fy*gx))/((bx*cy - by*cx - bx*ey + by*ex + \\ & cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)) + (Tg*(bx*cx*ey*fy - \\ & bx*cy*ex*fy - by*cx*ey*fx + by*cy*ex*fx - bx*cx*ey*hy + bx*cy*ex*hy + by*cx*ey*hx - \\ & by*cy*ex*hx - bx*cy*fx*hy + bx*cy*fy*hx + by*cx*fx*hy - by*cx*fy*hx + bx*ey*fx*hy - \\ & bx*ey*fy*hx - by*ex*fx*hy + by*ex*fy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - \\ & cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)) - (Tf*(bx*cx*ey*gy - bx*cy*ex*gy - \\ & by*cx*ey*gx + by*cy*ex*gx - bx*cx*ey*hy + bx*cy*ex*hy + by*cx*ey*hx - by*cy*ex*hx - \\ & bx*cy*gx*hy + bx*cy*gy*hx + by*cx*gx*hy - by*cx*gy*hx + bx*ey*gx*hy - bx*ey*gy*hx - \\ & by*ex*gx*hy + by*ex*gy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - \\ & fy*gx - fx*hy + fy*hx + gx*hy - gy*hx)) \end{aligned}$$

T2 =

$$\begin{aligned} & (Tf*(cy*dx + cx*ey - cy*ex - dx*ey)*(bx*gy - by*gx - bx*hy + by*hx + gx*hy - \\ & gy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + \\ & gx*hy - gy*hx)) - (Tg*(cy*dx + cx*ey - cy*ex - dx*ey)*(bx*fy - by*fx - bx*hy + by*hx + fx*hy \\ & - fy*hx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + \\ & gx*hy - gy*hx)) - (Th*(cy*dx + cx*ey - cy*ex - dx*ey)*(bx*fy - by*fx - bx*gy + by*gx + fx*gy \\ & - fy*gx))/((bx*cy - by*cx - bx*ey + by*ex + cx*ey - cy*ex)*(fx*gy - fy*gx - fx*hy + fy*hx + \\ & gx*hy - gy*hx)) \end{aligned}$$