



UNIVERSIDAD
DE PIURA

REPOSITORIO INSTITUCIONAL
PIRHUA

DISEÑO, SIMULACIÓN Y CONTROL DE UN ASISTENTE ROBÓTICO PARA CIRUGÍA LAPAROSCÓPICA

Manuel Amaya-González

Piura, mayo de 2017

FACULTAD DE INGENIERÍA

Máster en Ingeniería Mecánico-Eléctrica con Mención en Automática y
Optimización

Amaya, M. (2017). *Diseño, simulación y control de un asistente robótico para cirugía laparoscópica* (Tesis de Máster en Ingeniería Mecánico-Eléctrica con Mención en Automática y Optimización). Universidad de Piura. Facultad de Ingeniería. Piura, Perú.



Esta obra está bajo una licencia

[Creative Commons Atribución-NoComercial-SinDerivar 4.0 Internacional](https://creativecommons.org/licenses/by-nc-nd/4.0/)

[Repositorio institucional PIRHUA – Universidad de Piura](https://repositorio.institucional.pirhua.edu.pe/)

UNIVERSIDAD DE PIURA

FACULTAD DE INGENIERÍA



DISEÑO, SIMULACIÓN Y CONTROL DE UN ASISTENTE ROBOTICO PARA
CIRUGÍA LAPAROSCÓPICA

Tesis para optar el Grado de Máster en
Ingeniería Mecánico –Eléctrica con Mención en Automática y Optimización

Autor:

Bach. Manuel Amaya González

Asesores:

Dr. Ing. Carlos Jahel Ojeda Díaz

Dr. Ing. Edilberto Horacio Vásquez Díaz

Piura, Mayo 2017

*Dedico este trabajo de tesis
A mis padres, Manuel Isaías y María Esther, que con su esfuerzo me han ayudado
a alcanzar mis metas.
A mi hijo Liam, quien ha sido mi principal fuente de inspiración.*

Con todo mi amor,

Manuel

|

PRÓLOGO

La cirugía laparoscópica o mínimamente invasiva es un procedimiento quirúrgico en el cual se hacen pequeñas incisiones al paciente y a través de ellos se inserta el instrumental (pinzas, laparoscopio) según sea requerido. La cirugía robótica implica el uso de robots para su realización, y su desarrollo se debe a principalmente a que produce un incremento en la precisión y ergonomía del médico cirujano.

En el Perú el desarrollo de tecnología así como el uso de la misma, es bastante escasa en este ámbito. Si bien existen plataformas líderes en el rubro como el robot Da Vinci, son muy poco conocidas y su utilización es casi nula por los médicos peruanos. Además, cabe señalar que las cirugías laparoscópicas requieren de un médico asistente dedicado exclusivamente al manejo del laparoscopio o cámara (que transmite la imagen al monitor).

La principal motivación para el desarrollo de esta tesis es la de crear un prototipo propio que permita manipular autónomamente el laparoscopio, permitiéndole así al médico asistente realizar otras tareas complementarias a la operación. Por esta razón se hace el estudio y la simulación de un asistente robótico, así como la selección de motores comerciales que permitan su implementación. Además, la puesta en marcha de un prototipo piloto es posible, ya que en Piura existe un centro de entrenamiento para cirugía laparoscópica donde se hacen ensayos en pollos y cerdos.

Finalmente quiero agradecer a las personas que me apoyaron en la realización de este trabajo. Agradezco al Dr. Carlos Ojeda por su interés y motivación en la revisión de esta tesis. Al Dr. Mario Arbulú por su amabilidad y asesoría a pesar de la distancia. A mi padre el Sr. Manuel Amaya, que a través de CEPCEA (Centro de Estudios para la Prevención y Corrección de Enfermedades Abdominales) me abrió las puertas de las instalaciones y me presentó su realidad problemática. Al Dr. Edmundo Dediós por su colaboración al permitirme estar presente en sala de operaciones en una cirugía real (colecistectomía laparoscópica) dentro de la Clínica San Miguel – Piura. Por último una mención especial al CONCYTEC por el financiamiento para llevar a cabo estos estudios y la firme apuesta por la investigación en nuestro país.

RESUMEN

El objetivo del presente trabajo de investigación es el de realizar el estudio para el diseño, simulación y control de un brazo robótico que permita manipular la cámara o laparoscopio dentro de una cirugía laparoscópica. Esto constituye el primer paso en la intención de implementar un prototipo real que pueda ser probado inicialmente en un centro de entrenamiento de cirugía laparoscópica existente en nuestra ciudad.

El estudio inicia con la propuesta del diseño mecánico del robot. Se decide el número de grados de libertad, las dimensiones y el material de los eslabones. Con estos datos se realiza el modelo 3D y el modelamiento matemático (geométrico y cinemático) para relacionar el movimiento dentro del abdomen con el movimiento de las articulaciones. Se propone una trayectoria esférica a seguir dentro del abdomen y se realiza el modelamiento dinámico para calcular los torques necesarios para realizar dichos movimientos.

Finalmente con los torques calculados, se procede a seleccionar motores comerciales capaces de entregar dichos pares y con los datos provistos por el fabricante se modela y se aplica la técnica de control PID a dichos motores. Además, para simular el comportamiento del robot se exporta el modelo CAD al entorno de Matlab con SimMechanics, agregándole sensores y actuadores para seguir la trayectoria creada previamente.

Índice

| | |
|---|------|
| Prólogo | iii |
| Resumen | v |
| Índice de Figuras | xiii |
| Índice de Tablas..... | xvii |
| Introducción..... | 1 |
| | |
| Capítulo 1. Metodología de la investigación | 3 |
| 1.1 Planteamiento del problema..... | 3 |
| 1.2 Antecedentes | 4 |
| 1.2.1 En el mundo..... | 4 |
| 1.2.2 En el Perú | 6 |
| 1.2.3 Estructuras Funcionales de Robots..... | 8 |
| 1.2.3.1 Da Vinci | 8 |
| 1.2.3.2 Lapman | 9 |
| 1.2.3.3 Zeus | 10 |
| 1.2.3.4 Lapbot | 11 |
| 1.3 Objetivos..... | 11 |
| 1.3.1 Objetivos Generales..... | 11 |
| 1.3.2 Objetivos Específicos | 12 |
| 1.4 Justificación e Importancia | 12 |
| | |
| Capítulo 2. Cirugía Laparoscópica | 15 |
| 2.1 Cirugía mínimamente invasiva | 15 |
| 2.1.1 Definición | 15 |

| | | |
|---------------------------------------|--|----|
| 2.1.2 | Procedimiento | 16 |
| a) | Incisión..... | 16 |
| b) | Bióxido de Carbono | 17 |
| c) | Pinzas, Tijeras y Suturas | 17 |
| d) | Pos – Operatorio | 18 |
| 2.1.3 | Ventajas y desventajas de la CMI | 18 |
| 2.2 | Equipo e instrumental laparoscópico..... | 20 |
| 2.2.1 | La torre de video - laparoscopia..... | 20 |
| 2.2.1.1 | Laparoscopio u Óptica..... | 21 |
| 2.2.1.2 | Cámara..... | 23 |
| 2.2.1.3 | Fuente de luz y Cable de Fibra Óptica | 24 |
| 2.2.2 | Trócares..... | 24 |
| 2.2.3 | Equipo de insuflación..... | 25 |
| 2.2.4 | Instrumental quirúrgico | 27 |
| 2.3 | Distribución y posición del paciente | 28 |
| 2.3.1 | Mesa quirúrgica o camilla..... | 29 |
| Capítulo 3. Fundamentos teóricos..... | | 31 |
| 3.1 | Tipos de articulaciones | 31 |
| 3.2 | Cadena cinemática..... | 32 |
| 3.3 | Grados de libertad..... | 32 |
| 3.4 | Espacio Cartesiano | 33 |
| 3.5 | Transformación Homogénea | 33 |
| 3.5.1 | Parámetros Khalil – Kleinfinger | 34 |
| 3.5.2 | Parámetros Denavit - Hartenberg..... | 35 |
| 3.6 | Algoritmo de Denavit Hartenberg | 36 |
| 3.7 | Modelamiento | 36 |
| 3.7.1 | Modelo geométrico directo (MGD): | 37 |
| 3.7.2 | Modelo geométrico inverso (MGI) | 37 |
| 3.7.2.1 | Métodos Numéricos: Método de la Bisección..... | 38 |
| 3.7.3 | Modelo Cinemático Directo e Inverso | 39 |
| 3.7.3.1 | Jacobiana Analítica..... | 39 |
| 3.7.3.2 | Jacobiana Geométrica..... | 40 |
| 3.7.4 | Modelo Dinámico Directo e Inverso | 41 |

| | | |
|--|---|----|
| 3.8 | Control | 41 |
| 3.8.1 | Motor DC..... | 42 |
| 3.9 | Simulación | 43 |
| 3.9.1 | SolidWorks | 43 |
| 3.9.2 | SimMechanics | 43 |
| Capítulo 4. Diseño y Modelamiento Del Asistente Robótico | | 45 |
| 4.1 | Diseño IMEbot..... | 45 |
| 4.2 | Diseño 3D IMEbot..... | 47 |
| 4.3 | Modelamiento Geométrico IMEbot..... | 49 |
| 4.3.1 | Modelamiento Geométrico Directo (MGD) | 50 |
| 4.3.1.1 | Parámetros Khalil – Kleinfinger | 50 |
| 4.3.1.2 | Parámetros Denavit - Hartenberg..... | 53 |
| 4.3.2 | Modelamiento Geométrico Inverso (MGI)..... | 56 |
| 4.3.2.1 | Método de Paul | 57 |
| 4.3.2.2 | Desacoplo Cinemático | 64 |
| 4.3.1 | Prueba del Modelo Geométrico..... | 68 |
| 4.4 | Generación de Trayectoria..... | 70 |
| 4.5 | Modelamiento Cinemático IMEbot | 71 |
| 4.6 | Modelamiento Dinámico IMEbot..... | 75 |
| 4.6.1 | Parámetros inerciales | 76 |
| 4.6.2 | Formulación de Euler-Lagrange | 78 |
| 4.6.2.1 | Velocidades articulares de un robot manipulador..... | 79 |
| 4.6.2.2 | Energía Cinética de un Robot Manipulador..... | 81 |
| 4.6.2.3 | Energía Potencial de un robot Manipulador | 82 |
| 4.6.2.4 | Ecuaciones de movimiento de un manipulador | 83 |
| 4.6.3 | Formulación de Newton – Euler | 84 |
| 4.6.3.1 | Cinemática de los eslabones | 84 |
| 4.6.3.2 | Ecuaciones recursivas del movimiento del manipulador..... | 87 |
| 4.6.4 | Dinámica del robot usando Toolbox Robótica | 90 |
| 4.6.5 | Prueba de los modelos dinámicos inversos | 91 |
| Capítulo 5. Simulación y Control Del Asistente Robótico..... | | 95 |
| 5.1 | Modelamiento del Motor DC..... | 96 |

| | | |
|-----------|---|-----|
| 5.2 | Brushless DC Motor | 99 |
| 5.3 | Selección de Motores..... | 101 |
| 5.4 | Motores DC Maxon Brushless | 103 |
| 5.5 | Control de Motores Brushless | 106 |
| 5.6 | Simulación (SIMMECHANICS)..... | 111 |
| 6 | Conclusiones | 115 |
| 7 | Bibliografía..... | 117 |
| 8 | Glosario | 121 |
| 9 | Anexos..... | 123 |
| 9.1 | Metodo de paul | 123 |
| 9.2 | Codigos Matlab..... | 130 |
| 9.2.1 | Modelamiento Geométrico Directo..... | 130 |
| 9.2.1.1 | Parámetros Khalil – Kleinfinger..... | 130 |
| 9.2.1.1.1 | Función MGD_Khalil | 130 |
| 9.2.1.1.2 | Función MTH_Khalil..... | 131 |
| 9.2.1.2 | Parámetros Denavit – Hartenberg..... | 132 |
| 9.2.1.2.1 | Función MGD_DH | 132 |
| 9.2.1.2.2 | Función MTH_DH..... | 133 |
| 9.2.2 | Modelamiento Geométrico Inverso..... | 134 |
| 9.2.2.1 | Parámetros Khalil – Kleinfinger..... | 134 |
| 9.2.2.1.1 | Función MGI_Khalil..... | 134 |
| 9.2.2.1.2 | Método de la Bisección..... | 136 |
| 9.2.2.2 | Parámetros Denavit - Hartenberg | 137 |
| 9.2.2.2.1 | Función MGI_DH..... | 137 |
| 9.2.3 | Trayectoria Esférica | 139 |
| 9.2.4 | Generación de Trayectoria | 140 |
| 9.2.4.1 | Función Interpolar | 140 |
| 9.2.4.2 | Función splin_cúbico..... | 141 |
| 9.2.5 | Modelamiento cinemático | 142 |
| 9.2.6 | Modelamiento Dinámico Inverso..... | 143 |
| 9.2.6.1 | Formulación Lagrangiana..... | 143 |
| 9.2.6.2 | Formulación Newton-Euler | 148 |
| 9.2.6.3 | Usando el Toolbox de Robótica de Peter Corke..... | 151 |

10 Planos 155

ÍNDICE DE FIGURAS

| | |
|--|----|
| Figura 1.1 Colectomía laparoscópica..... | 3 |
| Figura 1.2 Asistente médico manipulando el laparoscopio..... | 4 |
| Figura 1.3 CEPCEA Training Center..... | 7 |
| Figura 1.4 Equipo de entrenamiento – CEPCEA..... | 7 |
| Figura 1.5 Maqueta Abdomen - CEPCEA..... | 8 |
| Figura 1.6 Vista isométrica de la estación quirúrgica y consola Da Vinci..... | 8 |
| Figura 1.7 Vista lateral de uno de los brazos robóticos Da Vinci..... | 9 |
| Figura 1.8 Diagrama esquemático correspondiente a la figura 1.7..... | 9 |
| Figura 1.9 Principio de funcionamiento de Lapman..... | 10 |
| Figura 1.10 Vista isométrica del sistema ZEUS..... | 10 |
| Figura 1.11 Grados de libertad de uno de los brazos articulados ZEUS..... | 11 |
| Figura 1.12 Lapbot..... | 11 |
| Figura 2.1 Procedimiento CMI–Incisión..... | 17 |
| Figura 2.2 Procedimiento CMI – CO2..... | 17 |
| Figura 2.3 Procedimiento CMI – Pinzas y tijeras..... | 18 |
| Figura 2.4 Procedimiento CMI – Post Operatorio..... | 18 |
| Figura 2.5 Componentes del sistema de imagen usados en CMI..... | 20 |
| Figura 2.6 Lentes interiores de un endoscopio rígido..... | 21 |
| Figura 2.7 Ángulos ópticos disponibles..... | 22 |
| Figura 2.8 Laparoscopios de 30° y 45°..... | 22 |
| Figura 2.9 Cámara/Cabezal. Cámara con acoplador para el endoscopio..... | 23 |
| Figura 2.10 Unidad de control de la cámara y Fuente de luz..... | 23 |
| Figura 2.11 Cable de fibra óptica..... | 24 |
| Figura 2.12 Emisión de luz en módulo de entrenamiento de cirugía laparoscópica..... | 24 |
| Figura 2.13 Punta del trocar..... | 25 |
| Figura 2.14 Trócares..... | 25 |
| Figura 2.15 Creación del neumoperitoneo con la aguja de Veress..... | 26 |
| Figura 2.16 Aguja de Veress..... | 26 |
| Figura 2.17 ENDOFLATOR 50, insuflador de alto rendimiento..... | 27 |
| Figura 2.18 Material para apendicectomía laparoscópica..... | 27 |

| | |
|---|----|
| Figura 2.19 Posición del paciente en una colecistectomía laparoscópica..... | 28 |
| Figura 2.20 Posición del paciente y del equipo quirúrgico en una apendicectomía laparoscópica..... | 29 |
| Figura 2.21 Mesa quirúrgica AMTAI T800 LSK..... | 29 |
| Figura 2.22 AMTAI T800 LSK: Base electrohidráulica y Mando de control..... | 30 |
| Figura 3.1 Articulación prismática..... | 31 |
| Figura 3.2 Articulación revoluta o rotacional..... | 32 |
| Figura 3.3 Cadena cinemática abierta..... | 32 |
| Figura 3.4 Parámetros geométricos Khalil – Kleinfinger..... | 34 |
| Figura 3.5 Parámetros geométricos Denavit - Hartenberg..... | 35 |
| Figura 3.6 Diagrama de bloques de un sistema de control de robots..... | 42 |
| Figura 3.7 Principio de funcionamiento motor DC..... | 42 |
| Figura 3.8 Diseño mecánico en Solidworks y SimMechanics..... | 44 |
| Figura 4.1 Estructura cinemática IMEbot..... | 46 |
| Figura 4.2 Base; Eslabón 1; Eslabón 2..... | 47 |
| Figura 4.3 Eslabón 3; Eslabón 4..... | 48 |
| Figura 4.4 Eslabón 5; Eslabón 6..... | 48 |
| Figura 4.5 Eslabón 7 (laparoscopio)..... | 48 |
| Figura 4.6 Ensamble final (asistente robótico)..... | 49 |
| Figura 4.7 Sistema de coordenadas con parámetros Khalil - Kleinfinger..... | 51 |
| Figura 4.8 Sistema de coordenadas 1 y 2 con parámetros D-H..... | 53 |
| Figura 4.9 Sistema de coordenadas 3, 4, 5, 6 y 7 con parámetros D-H..... | 54 |
| Figura 4.10 Valores numéricos con parámetros Denavit - Hartenberg..... | 55 |
| Figura 4.11 IMEbot montado en sala de operaciones 1..... | 56 |
| Figura 4.12 IMEbot montado en sala de operaciones 2..... | 57 |
| Figura 4.13 Movimiento alrededor del punto de incisión..... | 57 |
| Figura 4.14 Cálculo del MGI mediante Método de Paul..... | 64 |
| Figura 4.15 Desacoplo cinemático IMEbot..... | 65 |
| Figura 4.16 Cálculo del MGI mediante desacoplo cinemático..... | 67 |
| Figura 4.17 Trayectoria esférica..... | 68 |
| Figura 4.18 Prueba Modelo Geométrico..... | 69 |
| Figura 4.19 Posiciones cartesianas para describir una trayectoria esférica..... | 69 |
| Figura 4.20 Posiciones articulares para describir una trayectoria esférica..... | 70 |
| Figura 4.21 Error de posición por el uso de los modelos geométricos..... | 70 |
| Figura 4.22 Prueba Modelo Cinemático..... | 73 |
| Figura 4.23 Velocidades lineales del laparoscopio [m/s]..... | 74 |
| Figura 4.24 Velocidades angulares del laparoscopio [°/s]..... | 74 |
| Figura 4.25 Asignación de material en la pieza..... | 76 |
| Figura 4.26 Asignación de sistemas de coordenadas D-H..... | 77 |
| Figura 4.27 Parámetros inerciales..... | 77 |
| Figura 4.28 Ubicación de un punto arbitrario O_{ri} sobre el eslabón i | 79 |
| Figura 4.29 Sistema de coordenadas giratorio..... | 84 |
| Figura 4.30 Sistema de coordenadas en rotación y traslación..... | 85 |
| Figura 4.31 Relación entre los sistemas O, O^* y O' | 86 |

| | |
|---|-----|
| Figura 4.32 Fuerzas y momentos sobre eslabón i | 88 |
| Figura 4.33 IMEbot usando el Toolbox de Robótica | 90 |
| Figura 4.34 Prueba de modelos dinámicos | 91 |
| Figura 4.35 Error entre el método Lagrange y Newton-Euler..... | 91 |
| Figura 4.36 Error entre el método Newton-Euler y el proporcionado por el Toolbox de robótica Peter Corke | 92 |
| Figura 4.37 Torques en las articulaciones 1, 2 y 3 | 92 |
| Figura 4.38 Torques en las articulaciones 4, 5, 6 y 7 | 93 |
| Figura 5.1 Esquema eléctrico y mecánico de un motor dc..... | 96 |
| Figura 5.2 Diagrama esquemático de un Brushless DC Motor | 100 |
| Figura 5.3 Modelo a lazo abierto motor Maxon EC40..... | 105 |
| Figura 5.4 Diagrama de bloques Simulink motor Maxon EC40..... | 105 |
| Figura 5.5 Velocidad Motor Maxon EC 40 sin carga | 105 |
| Figura 5.6 Esquema de control PID | 106 |
| Figura 5.7 Control PID Motor Maxon EC22 con saturador (articulación 2) | 107 |
| Figura 5.8 Control PID Motor Maxon EC45 con saturador (articulación 3) | 107 |
| Figura 5.9 Control PID Motor Maxon EC40 con saturador (articulación 4) | 107 |
| Figura 5.10 Control PID Motor Maxon EC40 con saturador vs sin saturador..... | 108 |
| Figura 5.11 Control Motor Maxon EC max 22: PID vs PI vs PD..... | 109 |
| Figura 5.12 Control Motor Maxon EC 45: PID vs PI vs PD..... | 110 |
| Figura 5.13 Control Motor Maxon EC40: PID vs PI vs PD..... | 110 |
| Figura 5.14 Modelo importado de Solidworks..... | 111 |
| Figura 5.15 Bloque Machine Environment | 112 |
| Figura 5.16 Diagrama de bloques del robot IMEbot..... | 113 |
| Figura 5.17 Modelo SimMechanics incluyendo control de motores..... | 114 |
| Figura 5.18 Errores de posición | 114 |

ÍNDICE DE TABLAS

| | |
|---|-----|
| Tabla 1.1 Robots aplicados a cirugía..... | 6 |
| Tabla 2.1 Diferencias entre cirugía convencional y CMI | 19 |
| Tabla 3.1 Ecuaciones trigonométricas usadas en cinemática inversa..... | 37 |
| Tabla 4.1 Parámetros geométricos Khalil – Kleinfinger | 50 |
| Tabla 4.2 Parámetros geométricos Denavit - Hartenberg..... | 53 |
| Tabla 4.3 Parámetros inerciales: masa y densidad | 77 |
| Tabla 4.4 Parámetros inerciales: centro de gravedad y tensor de inercias | 78 |
| Tabla 5.1 Motor BLDC vs motor DC convencional | 99 |
| Tabla 5.2 Torques RMS y velocidades máximas | 101 |
| Tabla 5.3 Motores DC Brushless seleccionados. | 102 |
| Tabla 5.4 Motor Maxon EC-max 22..... | 103 |
| Tabla 5.5 Motor Maxon EC 45..... | 103 |
| Tabla 5.6 Motor Maxon EC 40..... | 104 |
| Tabla 5.7 Parámetros del controlador Motor Maxon EC max 22..... | 108 |
| Tabla 5.8 Parámetros del controlador Moto Maxon EC40..... | 108 |
| Tabla 5.9 Parámetros del controlador Motor Maxon EC 45 | 109 |

INTRODUCCIÓN

En el Perú una cirugía laparoscópica normalmente es desarrollada por dos médicos: el médico principal encargado de la disección, sutura y otros procedimientos, y un médico asistente encargado de manipular el laparoscopio. Este último debe mantener fija la imagen a través del laparoscopio y en muchas ocasiones debido a la duración de la intervención presenta ciertas fallas por la vibración de las manos o cediendo la posición del laparoscopio ocasionando que choque internamente en el abdomen y se produzca empañamiento del lente. Esto repercute en la calidad de una imagen óptima y estable, y no permite al médico asistente realizar otras actividades durante la intervención.

Estudios previos ^[1] han demostrado la utilidad de asistentes robóticos en cirugía. Así que con el propósito de desarrollar un prototipo propio y que pueda ser puesto en marcha en nuestra ciudad, se ha realizado el presente trabajo de investigación, el cual se ha estructurado de la siguiente manera:

El capítulo 1 contiene la parte introductoria, la cual presenta la realidad problemática, los robots empleados actualmente en cirugía y sus estructuras funcionales, así como también los objetivos que se pretende alcanzar, y la justificación e importancia del presente trabajo.

El capítulo 2 presenta los conceptos relacionados exclusivamente con la cirugía laparoscópica. Su definición, procedimiento y ventajas respecto a la cirugía tradicional. Además, se describe el equipo, instrumental quirúrgico y camilla empleados, y la distribución del paciente de acuerdo al tipo de intervención.

El capítulo 3 expone los fundamentos teóricos empleados en el campo de la robótica: tipos de articulaciones, grados de libertad, transformación homogénea, algoritmo Denavit – Hartenberg. También se explica el modelamiento geométrico, cinemático y dinámico de un robot. Finalmente se mencionan conceptos de control y softwares utilizados en la simulación

En el capítulo 4 se muestra el modelo CAD desarrollado. Se calcula el modelo geométrico (parámetros Khalil-Kleinfinger, Denavit-Hartenberg), el modelo cinemático y el modelo dinámico (formulación de Lagrange, Newton-Euler y usando una Toolbox de robótica en Matlab).

En el capítulo 5 se muestra el modelamiento del motor dc sin escobillas, la selección de los motores comerciales y su posterior simulación y control en Matlab. Por ultimo acoplamos estos modelos al modelo importado en SimMechanics del diseño CAD y simulamos el seguimiento de la trayectoria esférica creada.

Capítulo 1

Metodología de la Investigación

1.1 PLANTEAMIENTO DEL PROBLEMA

Idealmente, el médico cirujano debería tener control total de todos los instrumentos que son directamente requeridos para llevar a cabo un determinado procedimiento. Esto incluye los instrumentos quirúrgicos para la operación y el control del campo operatorio. Sin embargo, actualmente en el Perú las cirugías laparoscópicas, llamadas también mínimamente invasivas (CMI), se realizan con ayuda de un médico asistente, el cual, está encargado de manipular la cámara o laparoscopio durante toda la operación, mientras el médico principal realiza la cirugía.



Figura 1.1 Colecistectomía laparoscópica. A la derecha el médico principal (*Foto propia: Sala de Operaciones – Clínica San Miguel, Piura*)

Esta persona debido al prolongado tiempo que puede durar una intervención quirúrgica, puede presentar fallas a causa de fatiga, o ir perdiendo precisión en sus movimientos (vibración de las manos) repercutiendo en la recepción de una imagen óptima del interior del paciente. Además, esta persona debería tener la experiencia necesaria para

coordinar los movimientos del laparoscopio adecuadamente, de acuerdo a los requeridos en cada intervención quirúrgica.



Figura 1.2 Asistente médico manipulando el laparoscopio.
(Foto propia: Sala de Operaciones – Clínica San Miguel, Piura)

Kavoussi et al, en 1995 reportaron resultados de un estudio sobre la precisión y uso de un brazo robótico quirúrgico comparado con un asistente médico durante una cirugía laparoscópica urológica. Ellos concluyeron que el posicionamiento de la cámara era mucho más estable y con menores movimientos involuntarios en el brazo robótico en comparación al control humano. Además no encontraron diferencias significativas en los tiempos operativos entre uno u otro. [1]

La finalidad de diseñar un asistente robótico para la manipulación de la cámara es devolverle el control al cirujano principal, así como la entrega de una imagen estable y libre de temblores durante los procedimientos quirúrgicos, permitiendo al médico asistente realizar otras tareas complementarias durante la operación.

Por lo tanto, en este trabajo se pretende hacer el estudio que sienten las bases para la implementación de un primer prototipo de robot manipulador encargado del manejo del laparoscopio. Es decir, se debe diseñar, modelar, simular y controlar un brazo robótico para cirugía laparoscópica.

1.2 ANTECEDENTES

1.2.1 En el mundo

La evolución de la terapéutica quirúrgica ha sido vertiginosa y desde los principios clásicos de la cirugía abierta, la evolución ha llevado a través de la laparoscopia a la actual cirugía robótica, en la cual, se ha logrado la aplicación exitosa de los preceptos de ingeniería robótica al quehacer diario del cirujano. El desarrollo de la cirugía mínimamente invasiva,

inicialmente liderado por la laparoscopia convencional, constituyó terreno fértil para el desarrollo de técnicas quirúrgicas con el uso del robot. [2]

La introducción de innovaciones en cirugía es un proceso inherente al avance de la especialidad. La cirugía robótica ha introducido numerosas innovaciones en el campo de la cirugía mínimamente invasiva que han permitido implementar las habilidades y la precisión del cirujano. Inicialmente, la National Aeronautics and Space Administration (NASA) y la Defense Advanced Research Project (DARPA) desarrollaron los primeros robots aplicados a cirugía con el objetivo de reemplazar la presencia del cirujano en el campo de batalla y áreas catastróficas.

Posteriormente aparece el PUMA 200 para realizar biopsias cerebrales guiadas por tomografías computarizadas, el PROBOT para resecciones prostáticas guiadas por ultrasonidos y el sistema para cirugía ortopédica ROBODOC. Estos sistemas constituyen la primera generación de sistemas robóticos con configuración maestro-esclavo. [3]

Actualmente, existen dos tipos de sistemas robóticos: robots endoscópicos y robots telemanipulados. Cada sistema se diferencia en términos de la función que realiza. Los sistemas endoscópicos son útiles en la asistencia para manejo de la cámara durante el procedimiento quirúrgico (EndoAssist, AESOP, LapMan). Los robots telemanipulados o interfaces hombre-máquina (Maestro-Esclavo), son los sistemas robóticos completos (Zeus®, da Vinci®). Estas interfaces están conformadas por la tecnología más avanzada, sin embargo sus elevadas dimensiones y coste han dificultado su amplia difusión. [4]

A continuación se detalla brevemente los sistemas que han sido utilizados específicamente para la manipulación o sujeción del endoscopio (laparoscopia):

EndoAssist. Es un robot porta endoscopio controlado por los movimientos de la cabeza del cirujano, el cual, permite completa autonomía sobre los movimientos de la cámara. El movimiento se ejecuta por un emisor infrarrojo montado en la cabeza. El sensor está situado encima del monitor y reconoce cualquier orden ejecutado por éste. Además posee un “pedal” que asegura que no haya movimientos innecesarios cuando no se requiera. [5]

AESOP (Automated Endoscope System for Optimal Positioning). Es un robot que mantiene y mueve el laparoscopio durante la cirugía. El cirujano puede dirigir el brazo metálico articulado ya sea con un pedal de pie, control manual o usando comandos de voz. Con movimientos suaves del pie o de la mano puede cambiar de forma sencilla el laparoscopio en cualquier dirección. La presión aplicada por el cirujano controla la velocidad. [6]

LapMan. Es un porta laparoscopio dinámico guiado por un joystick fijado bajo el dedo índice del operador. Provee un control óptimo del campo visual mientras se está operando, asegurando estabilidad y desplazamientos suaves del laparoscopio. [7]

La siguiente tabla resume algunos robots que se han sido desarrollados:

| ROBOT | TIPO | APLICACIÓN | FABRICANTE |
|---------------------|------------------------------------|--|-----------------------------------|
| Padyc | <i>Restringido y algo autónomo</i> | <i>Cardiología</i> | <i>Schneider y otros</i> |
| Da Vinci | <i>Activo y teleguiado</i> | <i>Laparoscopia, Cardiología, Urología</i> | <i>Intuitive Surgical Sysrems</i> |
| Ler | <i>Activo y teleguiado</i> | <i>Endoscopia</i> | <i>Berkelman y otros</i> |
| AESOP | <i>Restringido y algo autónomo</i> | <i>Endoscopia</i> | <i>Intuitive Surgical System</i> |
| Endoassist | <i>Activo y teleguiado</i> | <i>Endoscopia</i> | <i>Armtrong Healthcare Inc</i> |
| Zeus | <i>Activo y teleguiado</i> | <i>Laparoscopia, Cardiología, Urología</i> | <i>Intuitive Surgical System</i> |
| Endobot | <i>Activo y algo autónomo</i> | <i>Laparoscopia</i> | <i>Kang y Wen</i> |
| Black Falcon | <i>Activo y teleguiado</i> | <i>Laparoscopia</i> | <i>Madhani</i> |

Tabla 1.1 Robots aplicados a cirugía ^[8].

1.2.2 En el Perú

A pesar del avance tecnológico actual y el uso de distintos robots para cirugía laparoscópica en otras partes del mundo como los mencionados anteriormente, en el Perú aún no se ha puesto en práctica el uso de asistentes robóticos. Esto puede deberse a la falta de inversión tanto estatal como privada, así como al cierto recelo de los médicos peruanos en el empleo de colaboradores robóticos en intervenciones quirúrgicas.

Sin embargo, ya se han realizado los primeros pasos para dar a conocer el uso y aplicación de estas nuevas tecnologías. En abril del 2014, EsSalud en colaboración con la Biblioteca Nacional del Perú invitaron al primer cirujano en aplicar tecnología robótica a pacientes con cáncer. El pionero en cirugía robótica a nivel mundial, Dr. David Samadi, llegó al Perú para la **Primera Conferencia Magistral: Robot Da Vinci-El Quirófano del Futuro Método Eficaz para el tratamiento del cáncer de próstata**. Esta incluía la capacitación a la comunidad científica peruana sobre el uso y aplicaciones del Robot Da Vinci en cirugías mínimamente invasivas, que a la fecha ya ha salvado la vida a más de 5,600 personas que padecen cáncer de próstata y otras neoplasias. ^[9]

En nuestra ciudad, Piura, existe un centro de entrenamiento para cirugía laparoscópica, CEPCEA Training Center. El Centro de Estudios para la Prevención y Corrección de Enfermedades Abdominales (CEPCEA) es una Organización No

Gubernamental, fundada en el 2002, que provee atención médica quirúrgica y además entrena a médicos cirujanos para perfeccionar sus técnicas en cirugía laparoscópica.

En el 2005 se dictó el primer curso de entrenamiento a 8 Cirujanos Generales en Colectectomía Laparoscópica. A mediados del 2009, con auspicio del Colegio Médico del Perú Filial Piura, se dictó el primer Diplomado en Cirugía Laparoscópica. Actualmente dicta cursos mensuales en Especialización de Cirugía Laparoscópica.



Figura 1.3 CEPCEA Training Center (*Foto propia*)



Figura 1.4 Equipo de entrenamiento – CEPCEA (*Foto propia*)

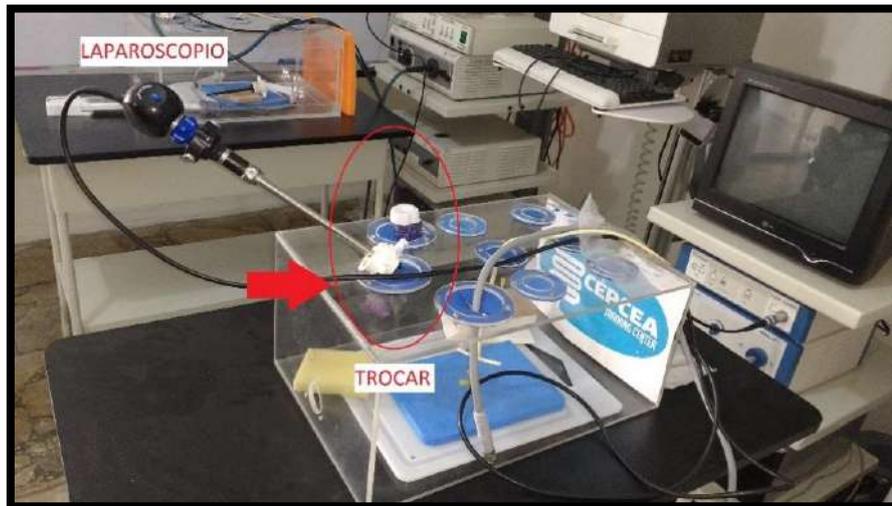


Figura 1.5 Maqueta Abdomen - CEPCEA (Foto propia)

1.2.3 Estructuras Funcionales de Robots

A continuación se presenta un breve resumen de las diferentes configuraciones estructurales usadas en los robots más importantes para aplicaciones en cirugía:

1.2.3.1 *Da Vinci*

El Sistema quirúrgico da Vinci es un sistema robótico manufacturado por Intuitive Surgical y diseñado para facilitar la compleja cirugía usando un enfoque mínimamente invasivo ^[10]. El sistema es controlado por un cirujano desde una consola, el cual posee 4 brazos interactivos. Dos de los brazos sostienen herramientas como tijeras o electrocauterios, el tercer brazo actúa como un bisturí. El cuarto brazo sostiene una cámara endoscópica que tiene dos lentes, que le da al cirujano una vista estereoscópica completa.

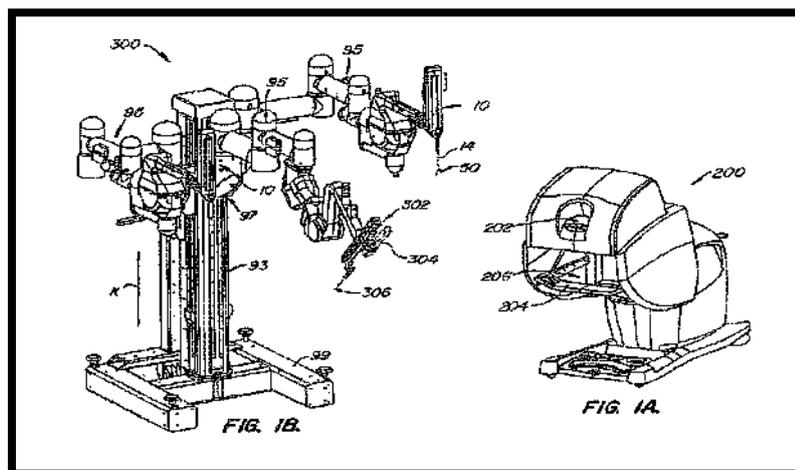


Figura 1.6 Vista isométrica de la estación quirúrgica y consola Da Vinci ^[11]

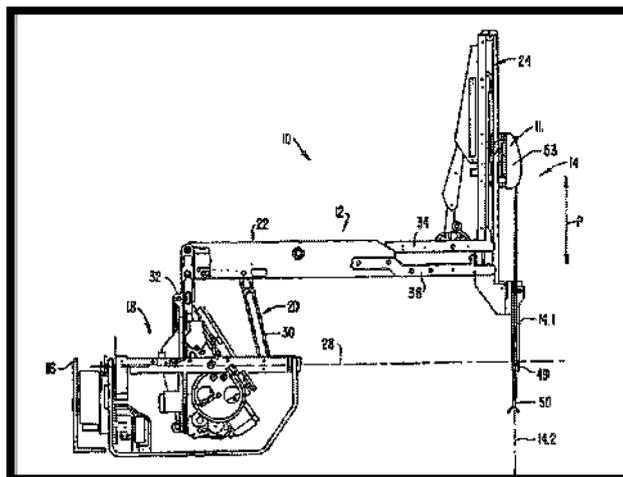


Figura 1.7 Vista lateral de uno de los brazos robóticos Da Vinci ^[11]

El sistema posee una cinemática cerrada de doble paralelogramo que le proporciona un centro de rotación, de tal forma que el robot siempre se posiciona a través del punto de incisión (49, en la figura 1.7).

Como se observa en la figura 1.8 (flechas 26, 27 y P) el brazo tiene 3 grados de libertad de movimiento.

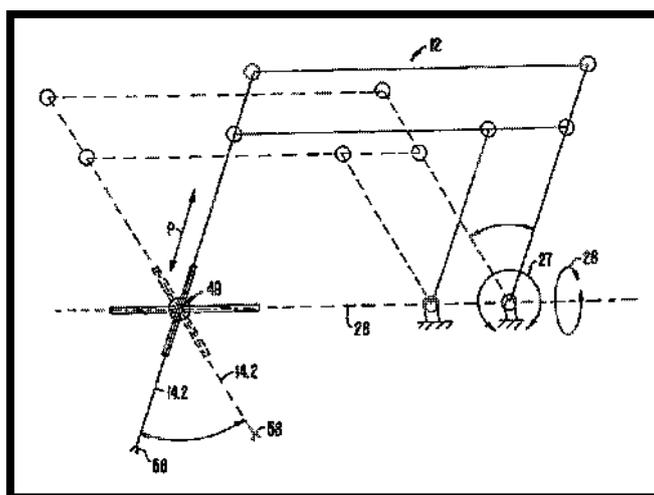


Figura 1.8 Diagrama esquemático correspondiente a la figura 1.7 ^[11]

1.2.3.2 Lapman ^[7]

La tecnología detrás de este manipulador está basada en el control electromecánico de unos frenos que regulan el desplazamiento de una serie de brazos articulados construidos para cubrir todo el espacio 3D. Lapman desplaza en los 3 ejes al laparoscopio que está conectado a él. Para tener óptimos movimiento de traslación, el centro geométrico debe estar alineado con el centro geométrico del paciente (el ombligo) a través de un puntero láser.

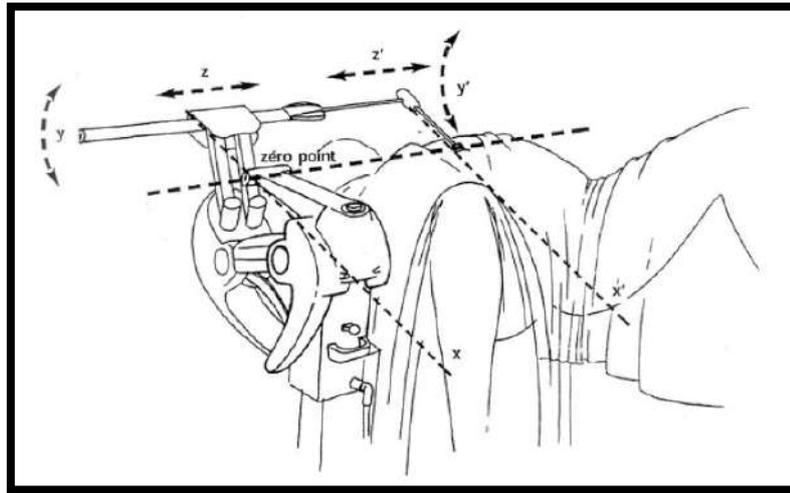


Figura 1.9 Principio de funcionamiento de Lapman ^[7]

1.2.3.3 Zeus ^[12]

Zeus Robotic Surgical System (ZRSS) producido originalmente por Computer Motion, consta de 3 brazos robóticos, los cuales eran controlados remotamente por el cirujano. El primero, AESOP, descrito anteriormente (20, en la figura 1.10), y los otros dos (16 y 18 en la figura 1.10) imitaban los movimientos del cirujano para hacer incisiones y extracciones más precisas.

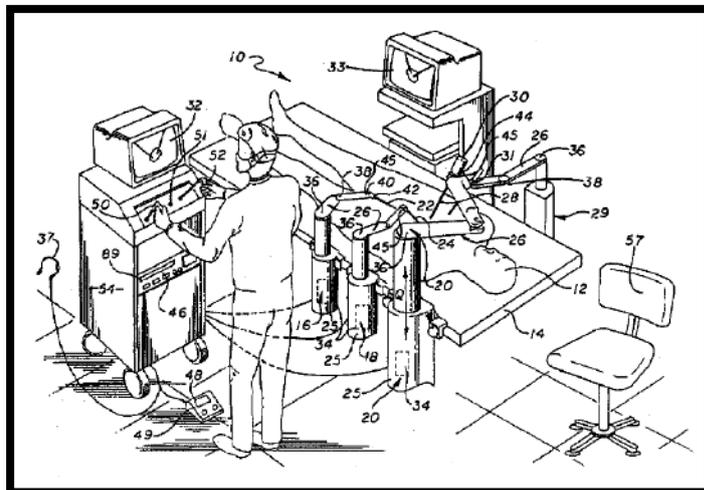


Figura 1.10 Vista isométrica del sistema ZEUS ^[12]

En la figura 1.11 se muestra los grados de libertad de uno de los dos brazos articulados para incisiones. Las articulaciones Js_1 , Js_2 y Js_3 son activas, mientras que Js_4 y Js_5 corresponden a juntas pasivas. La junta Js_6 rota los instrumentos quirúrgicos a lo largo de su eje. La junta Js_7 podría ser una pinza que abre y cierra. Los instrumentos se mueven tomando como punto de pivote P, que es la incisión del paciente.

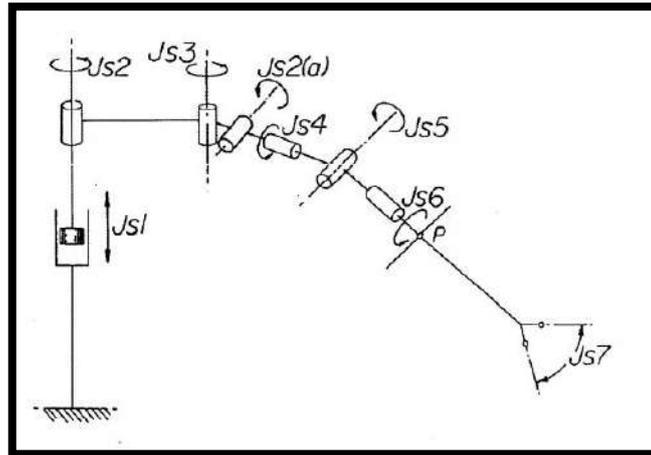


Figura 1.11 Grados de libertad de uno de los brazos articulados ZEUS ^[12]

1.2.3.4 Lapbot ^[13]

Proyecto desarrollado en la Universidad del Cauca, Colombia en el 2009 en el que se realiza la simulación de un robot para cirugía laparoscópica. Éste se basa en las estructuras anteriores y añade dos grados de libertad en la parte que va en el interior del paciente (después del paso del trocar) para poder manipular alguna pinza. El sistema posee 9 grados de libertad, de los cuales 3 son pasivos. Su estructura cinemática se muestra a continuación:

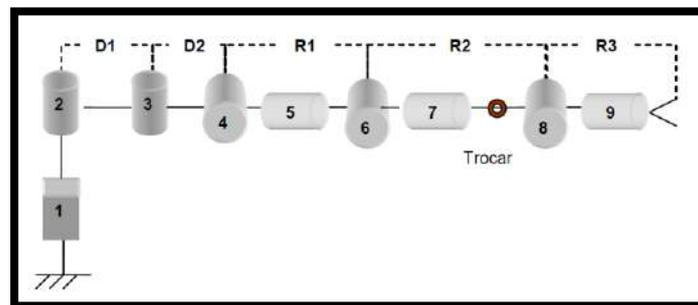


Figura 1.12 Lapbot

1.3 OBJETIVOS

1.3.1 Objetivos Generales

Diseñar, modelar y controlar un asistente robótico simulado (IMEbot) para cirugía laparoscópica que tenga como función principal la manipulación del laparoscopio (cámara), el cual podrá ser comandado directamente por el médico cirujano principal de acuerdo a sus requerimientos.

1.3.2 Objetivos Específicos

- Diseñar y seleccionar una estructura funcional óptima para el asistente robótico, teniendo en cuenta aspectos como sus grados de libertad, el tamaño, la estructura y ubicación del mismo dentro de una sala de operaciones.
- Modelar geoméricamente el asistente robótico IMEBot, tanto directa como inversa, usando parámetros definidos como Khalil – Kleinfinger y Denavit - Hartenberg y escoger el que tenga menor costo computacional, a fin de obtener sus valores numéricos en tiempo real. Se deberá tener en cuenta las restricciones de posición y movimientos.
- Modelar cinemáticamente el asistente robótico IMEBot, obteniendo la matriz Jacobiana que permita relacionar la velocidad de la posición y orientación del laparoscopio en función de sus velocidades articulares.
- Modelar dinámicamente el asistente robótico IMEBot, a fin de obtener las fuerzas o torques que se ejercerán en los actuadores para satisfacer las posiciones y velocidades articulares deseadas.
- Diseñar tridimensionalmente el asistente robótico mediante un software de simulación 3D como SOLIDWORKS, considerando los materiales de sus eslabones.
- Importar el diseño 3D del asistente robótico realizado en SOLIDWORKS al entorno SIMMECHANICS en Matlab, y aplicar técnicas de control adecuadas para establecer las posiciones y trayectorias deseadas, en el entorno de SIMULINK.

1.4 JUSTIFICACIÓN E IMPORTANCIA

El estudio del diseño y control de un asistente robótico mediante simulación para la manipulación del laparoscopio en cirugías laparoscópicas, es necesario como procedimiento previo a su implementación. La importancia de instaurar un robot que desempeñe esta función comúnmente realizada por un médico asistente se fundamenta principalmente en:

- Justificación Técnica, si el laparoscopio es guiado por un asistente robótico, se eliminaría el temblor de movimiento que posee un ser humano. Además se podría automatizar e incluso “aprender” movimientos de acuerdo al tipo de cirugía a efectuar. Esto generaría movimientos adecuados, que reducirían la cantidad de veces que el laparoscopio tiene que ser retirado para limpiarse debido a empañamiento.
- Justificación Económica, el uso de un asistente robótico, más allá del costo inicial que supone la adquisición del robot, reduciría los costos que implica los honorarios de un médico asistente calificado, ya que éste necesita un entrenamiento especial en la manipulación del laparoscopio. Además, a éste se le podría asignar otras labores dentro de la sala de operaciones.

- Justificación Ergonómica, el reemplazo del asistente médico por un robot supondría un mayor espacio de trabajo, es decir, un mayor rango libre para los movimientos del médico principal.
- Innovación, el uso de un robot dentro de una sala de operaciones en nuestra ciudad, mejoraría la percepción de nuestros médicos cirujanos por el uso de nuevas tecnologías, como los asistentes robóticos. Esto generaría un impacto social que fortalecería la relación medicina-ingeniería.
- Factibilidad de implementación, el desarrollo de un robot prototipo puede ser llevado a cabo teniendo como base el presente estudio. Una vez implementado, en Piura puede ponerse en fase piloto en el centro de entrenamiento de cirugía laparoscópica CEPCEA, en el cual los médicos realizan prácticas en pollos y cerdos.

Capítulo 2

Cirugía Laparoscópica

2.1 CIRUGÍA MINIMAMENTE INVASIVA

2.1.1 Definición

La cirugía mínimamente invasiva (CMI), también denominada de mínimo abordaje es el conjunto de técnicas diagnósticas y terapéuticas que por visión directa o endoscópica, o por otras técnicas de imagen, utiliza vías naturales o mínimos abordajes para introducir herramientas y actuar en diferentes partes del cuerpo. ^[14]

Es una técnica especializada para realizar que permite tener acceso a los órganos del paciente a través de pequeñas incisiones. La CMI se puede clasificar según el área del cuerpo a operar, así ^[15]:

- Laparoscopia: área abdominal o cavidad peritoneal.
- Toracoscopia: área torácica.
- Extracavitaria: en regiones cercanas al peritoneo y el tórax.

La cirugía laparoscópica usa varias incisiones de 0.5 a 1 cm. Cada incisión se denomina “puerto”. En cada puerto se inserta un instrumento tubular conocido como trocar. Durante el procedimiento, a través de los trocares se pasan instrumentos especializados y una cámara especial llamada laparoscopio.

Al iniciar el procedimiento, el abdomen se infla con el gas llamado dióxido de carbono para proporcionar al cirujano un espacio de trabajo y visibilidad. El laparoscopio transmite imágenes de la cavidad abdominal a los monitores de video de alta resolución del quirófano. Durante la operación, el cirujano observa las imágenes detalladas del abdomen en el monitor. El sistema permite que el cirujano realice las mismas operaciones que la cirugía tradicional pero con incisiones más pequeñas. ^[16]

Actualmente, los procedimientos quirúrgicos laparoscópicos más comunes son ^[8]:

- **Colecistectomía:** intervención que se realiza para quitar una vesícula biliar enferma, que está inflamada u obstruida por cálculos biliares.
- **Apendicectomía:** extirpación de un apéndice inflamado o infectado (apendicitis). Se hace una incisión pequeña en el cuadrante inferior derecho del abdomen y se extirpa el apéndice.
- **Histerectomía:** cirugía para extirpar el útero (matriz). El cirujano realiza una incisión por la cual corta las trompas de Falopio y separa el útero en su unión con el cuello.
- **Colectomía asistida:** extirpación o resección de una parte enferma del intestino grueso (colon).
- **Ligadura de trompas:** es una forma permanente de evitar el embarazo cerrando las trompas de Falopio de una mujer.
- **Gastrectomía:** extirpación parcial o total del estómago, en caso de presentarse problemas gástricos crónicos como úlceras o cáncer.
- **Prostatectomía:** intervención para extraer la totalidad o parte de la glándula prostática (próstata), para la curación del cáncer, la preservación de la continencia o la función sexual.
- **Hepatectomía:** procedimiento para diagnosticar, tratar o evacuar, las lesiones y tumores en el hígado.
- **Bypass gástrico:** Procedimiento para disminuir la capacidad del estómago y modificar la anatomía gastrointestinal con el fin de que la ingiera menos alimentos y disminuya la absorción de algunos nutrientes en el intestino, para así reducir o eliminar el sobrepeso.
- **Vasectomía:** procedimiento para lograr la esterilización masculina. Esta técnica consiste en cortar y ligar los conductos deferentes, impidiendo el recorrido del semen.

2.1.2 Procedimiento ^[17]

a) *Incisión*

La CMI es una técnica quirúrgica en la cual se insertan los trócares en el abdomen, utilizando pequeñas incisiones (inferiores a un centímetro). A través de éstos, se insertan instrumentos largos y angostos, que el cirujano utiliza para manipular, cortar y coser tejidos.

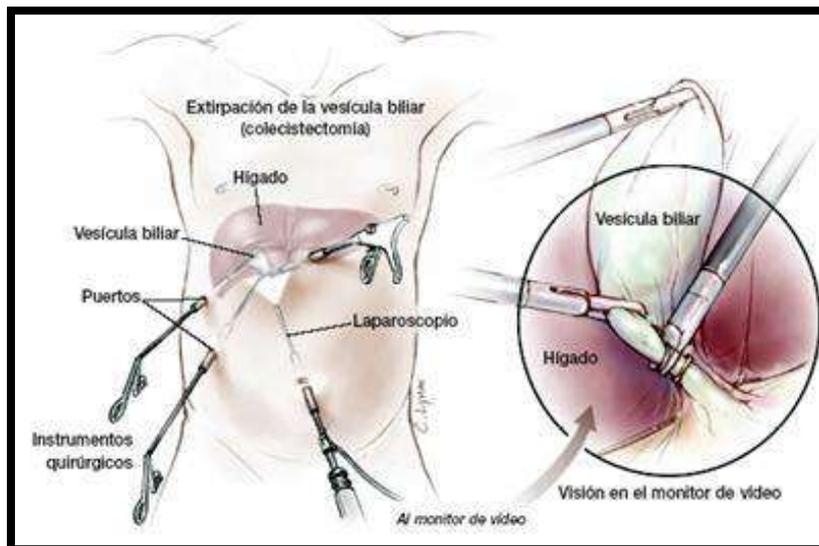


Figura 2.1 Procedimiento CMI–Incisión (<http://www.monografias.com/trabajos105/cirugia-minimamente-invasiva-robot-puma/image019.jpg>)

b) *Bióxido de Carbono*

Se introduce bióxido de carbono a través de uno de los trócares en el abdomen del paciente, elevando la pared abdominal y abriendo espacio para que el cirujano pueda trabajar. Con una cámara, insertada por medio de un trócar, la cual está conectada a un monitor de vídeo, permite al cirujano observar el contenido abdominal.

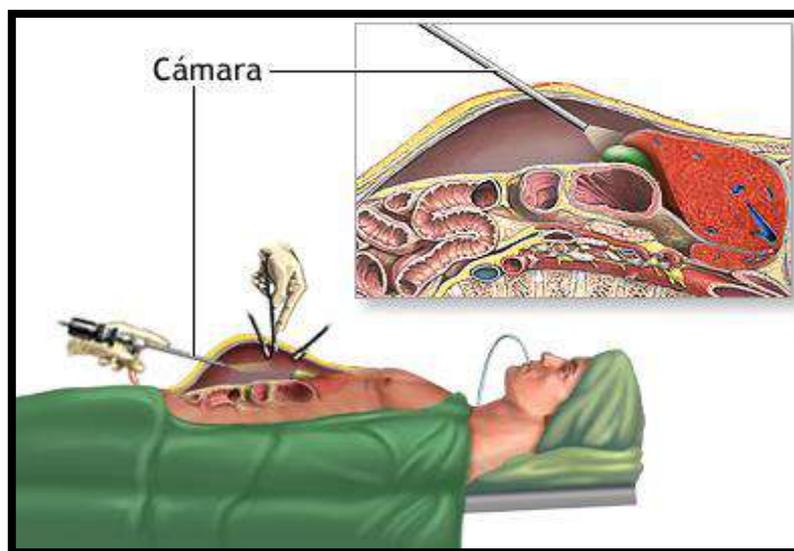


Figura 2.2 Procedimiento CMI – CO₂

c) *Pinzas, Tijeras y Suturas*

Las pinzas, las tijeras y las suturas que van insertadas al final de instrumentos largos y delgados pasan a través de los trócares.



Figura 2.3 Procedimiento CMI – Pinzas y tijeras

d) Pos – Operatorio

Por medio de la laparoscopia se pueden realizar una serie de procedimientos, como por ejemplo, la extirpación de la vesícula biliar (colecistectomía laparoscópica), la cirugía del esófago (funduplicatura laparoscópica), la cirugía del colon (colectomía laparoscópica) y la cirugía en el estómago y el bazo. Debido a que el cirujano hace sólo unas pequeñas incisiones, en vez de una incisión grande, el dolor postoperatorio es, por lo general, menor

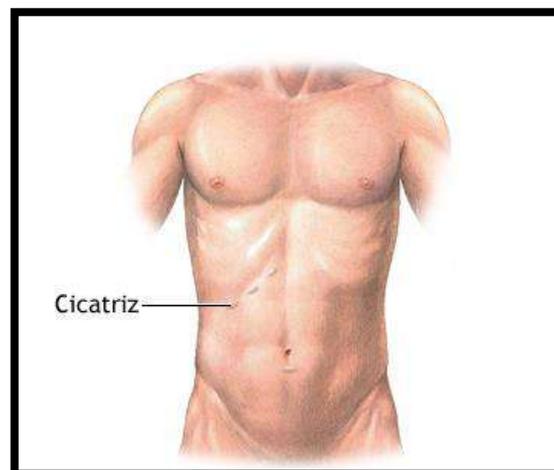


Figura 2.4 Procedimiento CMI – Post Operatorio

2.1.3 Ventajas y desventajas de la CMI

Con el desarrollo de las técnicas de cirugía mínimamente invasiva a finales de los 80's, los cirujanos ya no necesitaban colocar físicamente sus manos dentro del cuerpo para realizar una operación ^[15], desde entonces se han podido determinar algunas diferencias importantes entre la cirugía convencional y la cirugía mínimamente invasiva:

| <i>CIRUGÍA CONVENCIONAL</i> | <i>CIRUGÍA MINIMAMENTE INVASIVA</i> |
|--|--|
| <i>Basada en la anatomía</i> | <i>Basada también en simulación virtual</i> |
| <i>Interfaz sensorial directa</i> | <i>Interfaz video-endoscópica</i> |
| <i>Depende de la habilidad del cirujano</i> | <i>Depende también de la tecnología</i> |
| <i>Gran cantidad de herramientas</i> | <i>Cantidad limitada de herramientas</i> |
| <i>Mayor tiempo de recuperación del paciente</i> | <i>Menor tiempo de recuperación del paciente</i> |
| <i>Grandes marcas estéticas</i> | <i>Pequeñas marcas estéticas</i> |

Tabla 2.1 Diferencias entre cirugía convencional y CMI ^{[15], [18], [19]}.

La práctica de la Cirugía Mínimamente Invasiva presenta una serie de ventajas generales relacionadas con la técnica y aplicables a cualquier procedimiento. En general, todos los procedimientos se encuentran apoyados en la reducción de la morbilidad postoperatoria secundaria y en la disminución de la respuesta fisiológica a la agresión quirúrgica ^[14]:

- Reducción de la respuesta inflamatoria sistémica asociada con la cirugía, y mejoría en la respuesta inmunológica.
- Disminución del dolor postoperatorio debido fundamentalmente a la ausencia de incisiones quirúrgicas importantes y a la reducción del trauma en los tejidos sanos.
- Menores complicaciones en la herida quirúrgica. Las heridas tienen diámetros menores por lo que cicatrizan rápidamente y rara vez presentan complicaciones importantes. Además, es importante destacar el factor “estético” que conllevan estas mini incisiones.
- Disminución del postoperatorio y estancia en el hospital. En consecuencia, se reducen de forma significativa los costes asociados y las listas de espera.

Entre las principales desventajas podemos mencionar:

- Presenta dificultades con la percepción espacial: debido a que las intervenciones son controladas a través de monitores, se pierde la visión binocular que nos da la tridimensión. Sin embargo, gracias a los avances tecnológicos la tridimensión ya es una realidad.
- Pérdida de la percepción profunda (imposibilidad de palpación/sensación). La sensación y el tacto de la cirugía convencional se pierde y es necesario aprender a palpar con los instrumentos.
- Manejo oncológico discutible debido principalmente a la falta de resultados.
- En caso de sangrado importante, éste es difícil o imposible de controlar.
- Los nódulos pequeños y profundos son difíciles de ubicar.
- El proceso de suturación es más lento y complicado.

2.2 EQUIPO E INSTRUMENTAL LAPAROSCÓPICO

Se han diseñado múltiples instrumentos dirigidos a cubrir las demandas y necesidades de la laparoscopia. El equipamiento básico para la cirugía laparoscópica consta de [20]:

- a) La torre de video-laparoscopia
- b) Los trocares
- c) El equipo de insuflación
- d) El instrumental quirúrgico y
- e) El sistema de aspiración e irrigación.

2.2.1 La torre de video - laparoscopia

La cirugía mínimamente invasiva laparoscópica depende del funcionamiento colectivo de los componentes de imagen (figura 1.5). Un mal funcionamiento en cualquiera de los componentes afecta a los otros y puede reducir la seguridad del paciente [21].

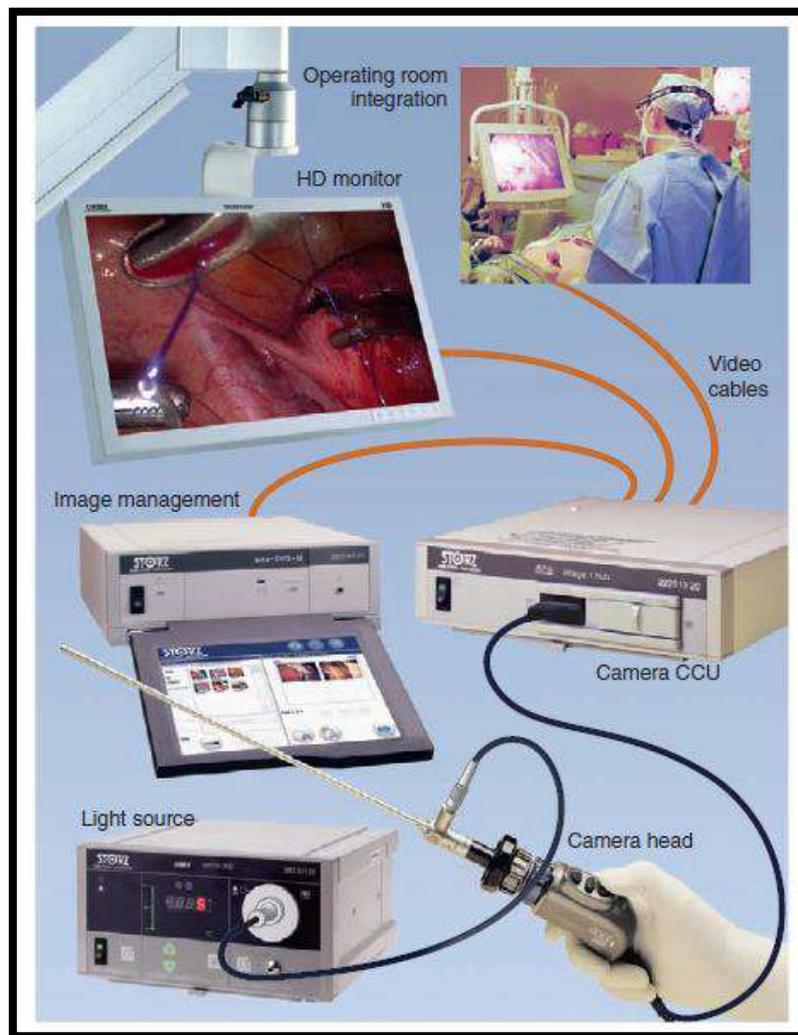


Figura 2.5 Componentes del sistema de imagen usados en CMI [21].

Consta de varias unidades: laparoscopio u óptica, la cámara (cabezal), el monitor de video y la fuente de luz. Antes de comenzar la cirugía hay que asegurarse de que todo el sistema óptico funciona adecuadamente y está en perfecto estado. Habitualmente en la torre va también colocado el equipo de insuflación [20].

La imagen empieza con el laparoscopio; la fuente de luz de xenón se conecta al laparoscopio por un cable de fibra óptica. La cámara/cabezal lee las imágenes y las transmite a la unidad de control de la cámara (UCC). Desde allí se transmiten las imágenes a una unidad de gestión de imágenes, la cual puede producir archivos digitales de las imágenes. La UCC también transmite imágenes digitales al monitor, las cuales son necesarias para ver el campo operatorio. Además la imagen se proporciona a otros monitores o sistemas de información a través de un sistema integrado de quirófano [21].

2.2.1.1 Laparoscopio u Óptica

El laparoscopio es un instrumento óptico alargado que se introduce a través de una cánula. La lente proximal del telescopio se llama **ocular** y la distal se llama **objetivo**. Contiene el sistema de lentes serial, que capta las imágenes iluminadas en la punta. La luz se transmite a través del cable de fibra óptica, que se une con el ocular.

Las características ópticas y dimensiones de los laparoscopios son:

- *Ángulo óptico*: dirección en la cual los lentes se enfocan en la imagen. Se mide en grados. Los más comunes son:
 - Frontal o de cero grados 0° , similar al ojo humano
 - Frontal oblicuo o de 30 grados.
 - De 45° .
 - De cero grados de 2 mm.
- *Diámetro*: diámetro del eje del telescopio, medido en milímetros (mm).
- *Longitud*: longitud del eje del telescopio, medido en milímetros (mm).

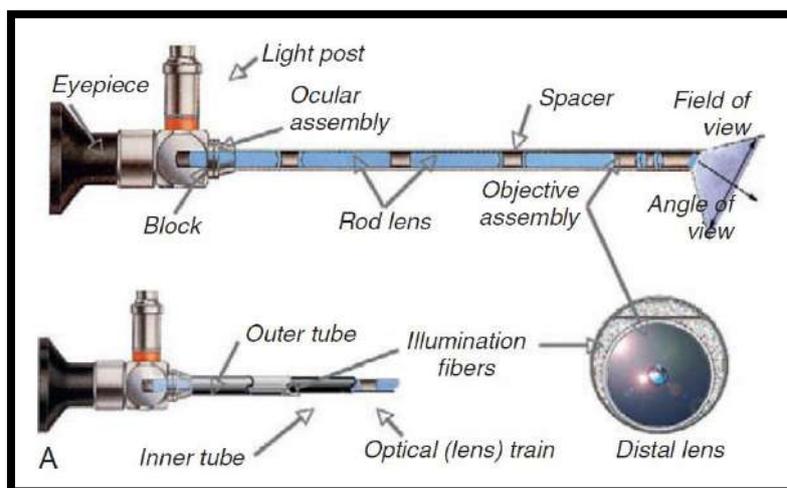


Figura 2.6 Lentes interiores de un endoscopio rígido [21].

En cirugía laparoscópica las más utilizadas son las de diámetro 10mm, bien de 0° o de 30° ^[5]. La longitud promedio es de 30 cm. Habitualmente es manejado por el primer o segundo ayudante y de su correcto manejo (acercarse, alejarse, guardar una adecuada distancia al lecho quirúrgico) depende en buena parte el éxito de la intervención.

Con frecuencia a lo largo de la cirugía se nos empañará o manchará. El empañamiento se produce por el depósito de restos tisulares o sangre sobre el objetivo y por la diferencia de temperatura entre el aire y el cuerpo. Pueden usarse intermitentemente agentes especiales desempañantes durante el procedimiento, pero las lentes deben enjuagarse y secarse antes de aplicar dicho agente.

Un método alternativo para desempañar son los calentadores de ópticas, pequeños termos que el instrumentista tiene en la mesa de instrumental para guardar suero caliente que permita limpiar la óptica por inmersión sin que esta pierda temperatura y vuelva a empañarse al entrar de nuevo en la cavidad abdominal.

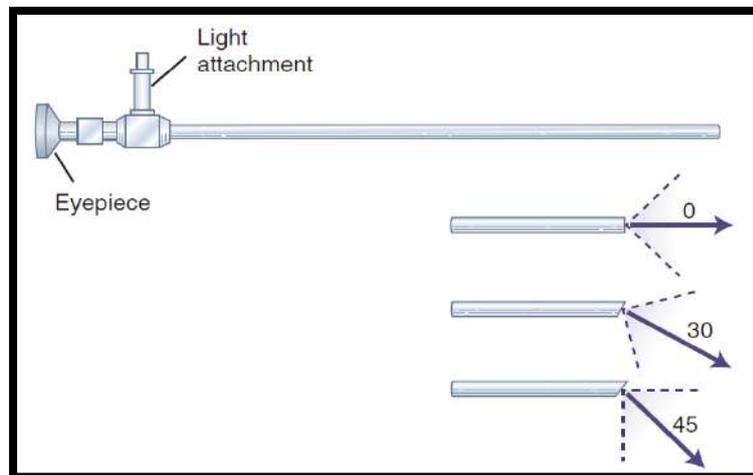


Figura 2.7 Ángulos ópticos disponibles ^[21].



Figura 2.8 Laparoscopios de 30° y 45° (Foto propia: CEPCEA, Piura).

2.2.1.2 Cámara

La cámara es un elemento primordial, de su calidad dependerá la mayor o menor percepción de profundidad, aspecto de gran importancia dada la falta de visión tridimensional. [20]

La calidad de la imagen depende de la calidad de los laparoscopios, la fuente de luz y los capacitores electrónicos del sistema. Las video cámaras modernas contienen 1 o 3 chips de silicio de estado sólido, los cuales producen señales eléctricas que son amplificadas y mostradas en un monitor.

Las cámaras con 3-chips producen imágenes de color natural, que son importantes en la identificación de patologías. Cada elemento de silicio en el chip representa un píxel. La resolución de la imagen depende de cuántos píxeles o unidades de silicio contenga el chip. [21]

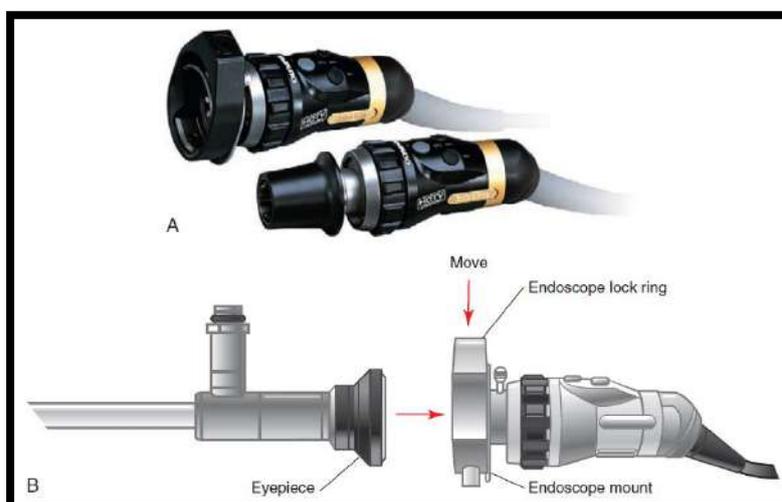


Figura 2.9 Cámara/Cabezal. Cámara con acoplador para el endoscopio [21].



Figura 2.10 Parte superior: Unidad de control de la cámara.
Parte inferior: Fuente de luz (Foto propia: CEPCEA, Piura).

2.2.1.3 Fuente de luz y Cable de Fibra Óptica

Es necesario disponer de una fuente de luz fría de intensidad regulable con suficiente potencia (350-500 Watt) para conseguir una adecuada luminosidad dentro de la cavidad abdominal. La luz se transmite por un cable de fibra óptica fijado al laparoscopio. El cable de fibra óptica está compuesto por múltiples fibras que están alineadas de forma paralela. La luz se proyecta a lo largo de las fibras del cable y se refleja en su interior. De esta manera la luz se concentra y puede dirigirse al flexionar el cable.



Figura 2.11 Cable de fibra óptica



Figura 2.12 Emisión de luz en módulo de entrenamiento de cirugía laparoscópica (*Foto propia: CEPCEA, Piura*).

2.2.2 Trócares

Son instrumentos diseñados para proporcionar el acceso a la cavidad abdominal y constituyen los canales de trabajo por los cuales se introducen la óptica y los distintos instrumentos. Están provistos de un sistema de válvulas que evitan la pérdida del neumoperitoneo y un canal de ingreso de CO₂ con llave de paso de una vía para mantenerlo [22].

La palabra “trocar” es usualmente utilizada para referirse al ensamblaje del aparato completo. Trocar realmente es el estilote que es introducido a través de la cánula. Los trocares están disponibles de diversas formas de acuerdo a su punta. Las puntas cortantes de estos trocares

son de diferentes formas: piramidal de 3 lados o cuchillas planas de 2 lados. Las cánulas son en realidad hechas de plásticos o de metales. Los mecanismos plásticos ya sean transparentes u opacos necesitan designársele de tal forma que disminuya la reflexión de la luz del telescopio. ^[23]

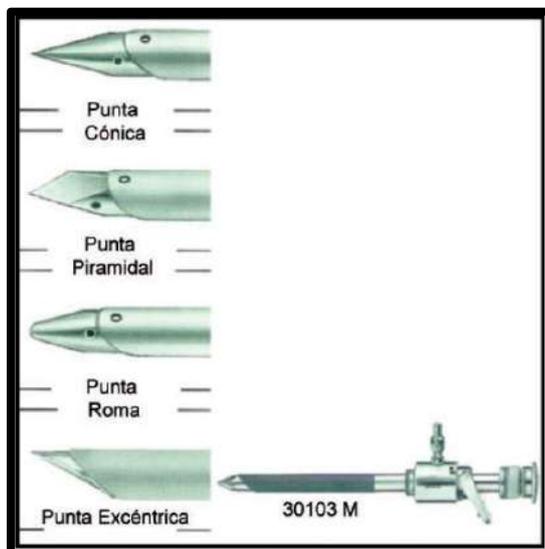


Figura 2.13 Punta del trocar ^[23]



Figura 2.14 Trócares (Foto propia: Sala de Operaciones – Clínica San Miguel, Piura)

2.2.3 Equipo de insuflación

Para acceder a la patología a tratar por laparoscopia, el cirujano debe crear una cavidad real en el abdomen. Esto se logra con la realización de un neumoperitoneo a través de la insuflación controlada de CO₂, el cual separa la pared abdominal y comprime el intestino, creando una burbuja de gas que permite la inserción del primer trocar y la creación del campo

quirúrgico. Este neumoperitoneo se realiza utilizando como **equipamiento** el *neumoinсуflador electrónico* y como **instrumental** la *aguja de Veress* [22].

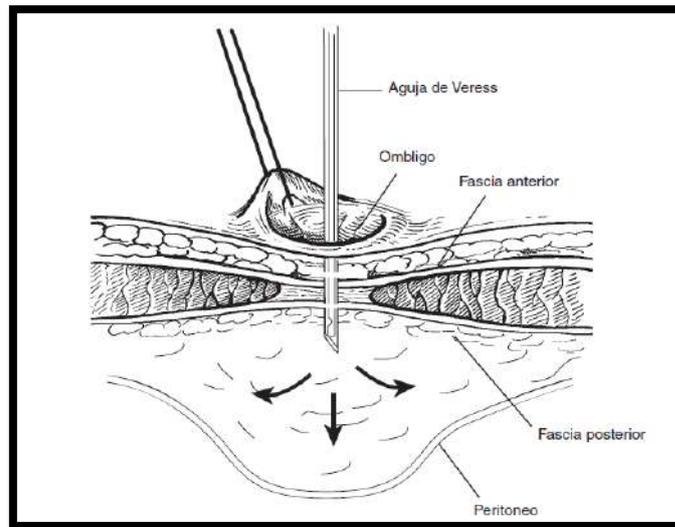


Figura 2.15 Creación del neumoperitoneo con la aguja de Veress [24].

Aguja de Veress.- La aguja de Veress consiste de una cánula externa que tiene una aguja puntiaguda para cortar a través de los tejidos. Dentro de la cánula existe un estilete interno que es cargado con un resorte y al momento de la disminución de la presión causada al atravesar la pared abdominal y al entrar en la cavidad peritoneal se inactiva. El cuerpo lateral de este estilete permite el paso del CO₂ hacia la cavidad intraabdominal.

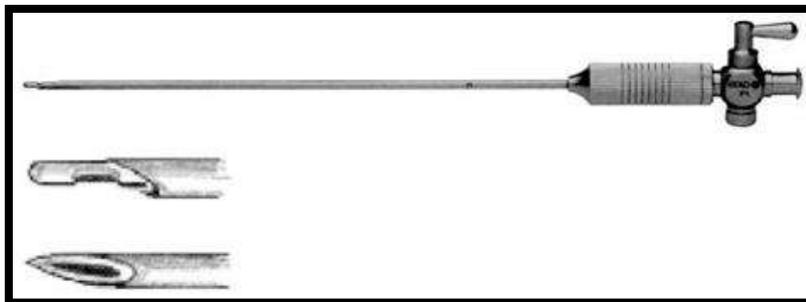


Figura 2.16 Aguja de Veress

(<http://marcelaparoscopia.blogspot.com/2016/03/sala-de-cirugia-preparar-el.html>)

Dióxido de Carbono.- Se usa CO₂ para la insuflación porque está rápidamente disponible, es estable y el cuerpo lo absorbe en forma natural. La incidencia de embolia con este gas es más baja que con el aire. No estimula la combustión y no tiene olor. [21]

Insuflador.- Es el aparato médico para introducir el dióxido de carbono. Este dispositivo mide la presión y el flujo de gas hacia el abdomen. Un insuflador debe tener como principales características el ajuste para flujos altos y bajos, capacidad de calentar el gas y compensar eficientemente las pérdidas. [21]



Figura 2.17 ENDOFLATOR 50, insuflador de alto rendimiento.

(https://www.karlstorz.com/cps/rde/xbcr/karlstorz_assets/ASSETS/3388607.pdf)

2.2.4 Instrumental quirúrgico

Los instrumentos usados en cirugía mínimamente invasiva son finos porque deben ingresar por incisiones pequeñas y actuar en el interior del cuerpo. Es delicado y requiere un mantenimiento cuidadoso ^[21]. El instrumental de uso estándar es de 5 y 10mm de diámetro y 35cm de longitud, y puede ser reutilizable o de un solo uso ^[20].

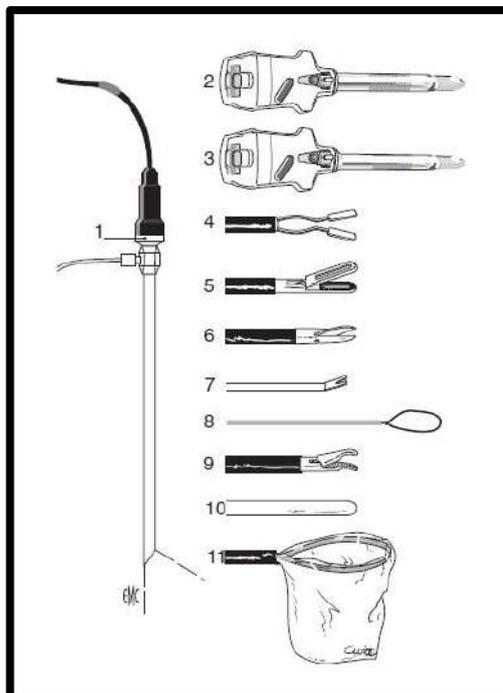


Figura 2.18 Material para apendicectomía laparoscópica ^[25]

1. Óptica de 30°;
2. Trocar de 10 mm;
3. Trocar y mandril romo de 5 mm;
4. Pinza bipolar coagulante de 5 mm;
5. Pinza atraumática de 5 mm;
6. Tijeras de 5 mm;
7. Empujanudos;
8. Nudo prefabricado;
9. Portaagujas;
10. Palpador romo;
11. Bolsa de recuperación para tejidos blandos.

Pueden estar recubiertos de material aislante para poder ser conectado a una fuente de diatermia mono o bipolar. Clasificación genérica ^[8]:

- Instrumentos de disección.
- Instrumentos de agarre.
- Instrumentos de ligadura y sutura.
- Instrumentos de punción y biopsia.

- Instrumentos de aspiración e irrigación.

2.3 DISTRIBUCIÓN Y POSICIÓN DEL PACIENTE

En el quirófano laparoscópico es fundamental la distribución adecuada de los diferentes elementos y los cables que posibilitan su uso para realizar el procedimiento con éxito.

Habitualmente en la *cirugía renal*, los cables de la cámara, luz y del insuflador que vienen de la torre entrarán en el campo quirúrgico por la cabecera del paciente, mientras que los cables que conectan el instrumental electroquirúrgico y el aspirador/irrigador con sus respectivos generadores llegan al campo desde los pies del enfermo.

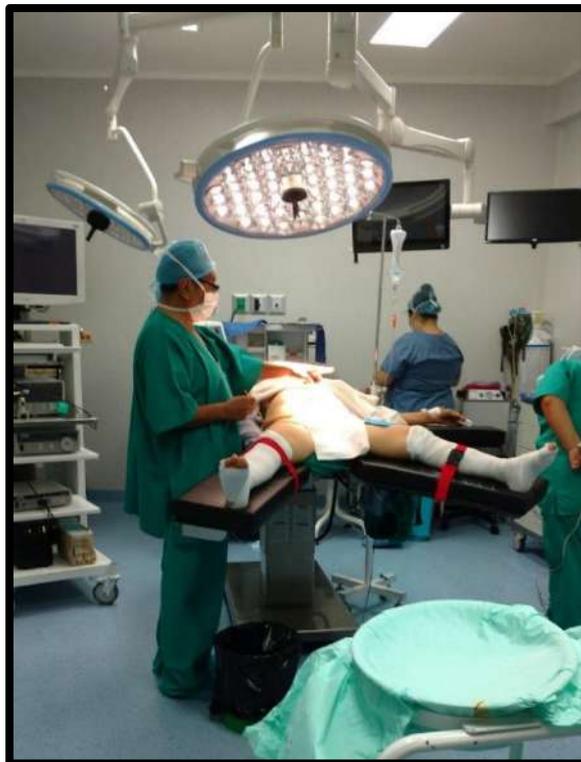


Figura 2.19 Posición del paciente en una colecistectomía laparoscópica (*Foto propia: Sala de Operaciones – Clínica San Miguel, Piura*)

En la *cirugía abdominal*, por ejemplo en una colecistectomía la torre junto con el monitor se coloca en la parte lateral derecha del paciente. El médico principal ingresa por los pies y el médico asistente, quien manipula el laparoscopio, se ubica en la parte lateral izquierda del paciente.

En la cirugía pélvica, la torre se colocará a los pies del paciente, con lo que todos los cables y conexiones entran por los pies del enfermo (la cámara, la luz y el insuflador entran por las piernas del paciente, el resto de cables al estar situados los generadores a ambos lados de la torre entrarán de la misma forma al campo quirúrgico).^[20]

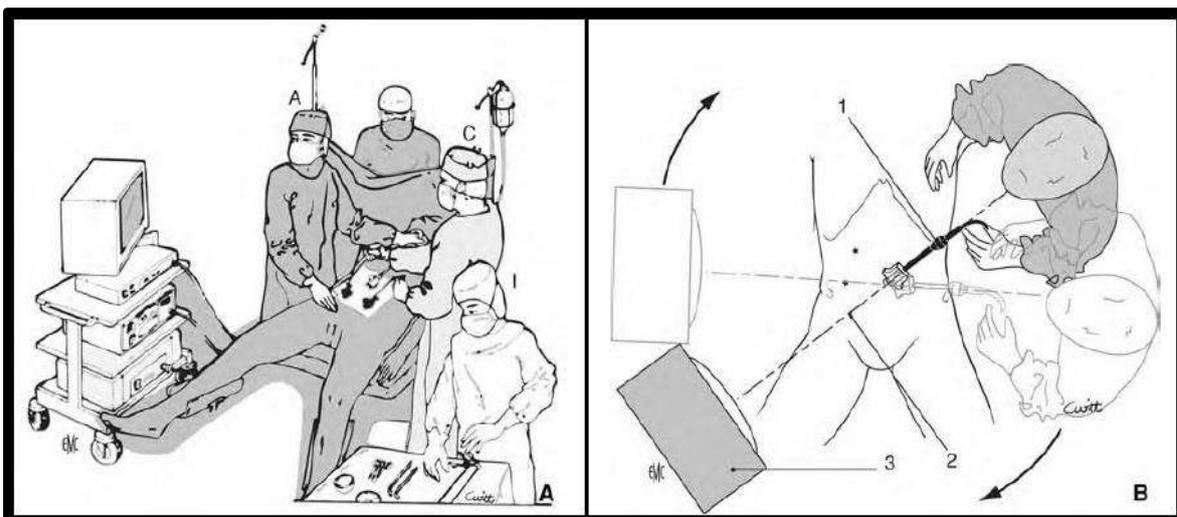


Figura 2.20 Izquierda: Posición del paciente y del equipo quirúrgico en una apendicectomía laparoscópica (A. ayudante; C. cirujano; I. instrumentista). Derecha: Eje cirujano – zona quirúrgica – monitor (1. Sistema óptico; 2. Eje de visión; 3. Monitor). [25]

2.3.1 Mesa quirúrgica o camilla

Una mesa quirúrgica es la herramienta que emplea el cirujano para el sostenimiento del paciente durante una intervención quirúrgica y deberá permitir realizar las maniobras quirúrgicas necesarias. Entre otras características debe ser estable y confortable, tener un acolchado adecuado y una base electrohidráulica.



Figura 2.21 Mesa quirúrgica AMTAI T800 LSK (Foto propia: Sala de Operaciones – Clínica San Miguel, Piura)

Posee un mando que regula sus movimientos [26]:

- Regulación de altura.
- Inclinación lateral a ambos lados.
- Trendelenburg y antitren.

- Regulación de la placa lumbar.
- Regulación de las piernas por separado y juntas.
- Regulación de la placa de la cabeza.



Figura 2.22 AMTAI T800 LSK Izquierda: Base electrohidráulica. Derecha: Mando de control (Foto propia: Sala de Operaciones – Clínica San Miguel, Piura)

Capítulo 3 Fundamentos Teóricos

3.1 TIPOS DE ARTICULACIONES

Según SAHA ^[27], un robot se compone de varios eslabones que se conectan en forma serial mediante articulaciones. Los eslabones se acoplan mediante juntas o articulaciones cinemáticas. Existen diferentes tipos de articulaciones. Las más utilizadas en robótica son la prismática y de revoluta o rotacional.

- **Prismática:**
Permite que 2 eslabones se deslicen uno respecto al otro, a lo largo de su eje. Impone 5 restricciones, por tanto tiene un grado de libertad (GDL).
- **Rotacional:**
Permite que 2 eslabones giren uno respecto al otro, alrededor del eje de articulación. Impone 5 restricciones, por tanto tiene un grado de libertad (GDL).

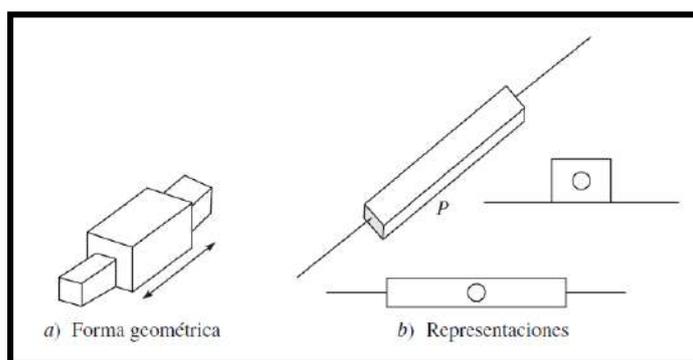


Figura 3.1 Articulación prismática ^[27]

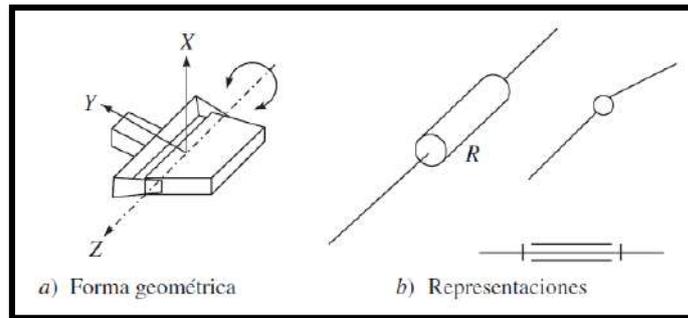


Figura 3.2 Articulación revolvente o rotacional [27]

3.2 CADENA CINEMÁTICA

Una cadena cinemática está formada por un conjunto de eslabones que están unidos mediante articulaciones. Las cadenas cinemáticas pueden ser abiertas o cerradas. Es abierta si los eslabones extremos sólo están unidos a un eslabón, y cerrada si están unidos a otros dos. Los eslabones se consideran como cuerpos rígidos y definen cómo se relacionan dos ejes de articulaciones consecutivas dentro de una cadena cinemática.

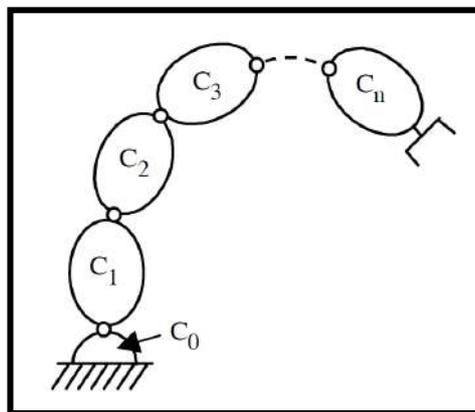


Figura 3.3 Cadena cinemática abierta [28]

3.3 GRADOS DE LIBERTAD

El grado de libertad de un robot es el número de coordenadas independientes o coordenadas mínimas necesarias para describir con certeza su posición o configuración. Es decir, un cuerpo rígido que se mueve en un espacio cartesiano 3D tendría 6 GDL, tres para su posición y tres para la orientación.

En CRAIG [29], se hace énfasis que por consideraciones de diseño mecánico se recomienda que los manipuladores se construyan con articulaciones que contengan sólo un grado de libertad.

3.4 ESPACIO CARTESIANO

Es la representación del espacio de trabajo del robot, dentro del cual podemos definir la ubicación, posición y orientación del efector final (laparoscopio o cámara). Donde iP es la posición de un punto arbitrario P con respecto al sistema de referencia R_i , mientras que la orientación de un sistema R_i a otro R_j se representa por la matriz de orientación iA_j .

$${}^iP = [{}^iP_x \quad {}^iP_y \quad {}^iP_z]^T \quad (3.1)$$

$${}^iA_j = [{}^i s_j \quad {}^i n_j \quad {}^i a_j] = \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix} \quad (3.2)$$

3.5 TRANSFORMACIÓN HOMOGÉNEA

En robótica, normalmente se asignan sistemas de coordenadas a cada eslabón del robot. La transformación entre sistemas nos permite:

- Calcular la ubicación, posición y orientación de los eslabones del robot relativos a cada uno de los otros.
- Describir la posición y orientación de los objetos
- Especificar trayectorias y la velocidad del efector final
- Describir y controlar las fuerzas cuando el robot interactúa con su entorno.

La transformación, translación y/ rotación de un sistema S_i a otro S_j se representa por iT_j :

$${}^iT_j = [{}^i n_j \quad {}^i o_j \quad {}^i a_j \quad {}^i P_j] = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Donde ${}^i n_j, {}^i o_j, {}^i a_j$ son las componentes de los vectores unitarios x_j, y_j, z_j , del sistema S_j expresados en términos del sistema S_i y ${}^i P_j$ es el vector representando las coordenadas del origen de S_j expresados en términos del sistema S_i . La matriz de transformación homogénea queda siempre bien definida por 4 grados de libertad o parámetros, si es que sus sistemas de coordenadas se han elegido correctamente. Estas 4 transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que componen la matriz de transformación homogénea, relacionan al sistema de referencia $i - 1$ con el sistema i ^[30]. Estos pueden ser los parámetros Khalil – Kleinfinger o los parámetros Denavit – Hartenberg.

3.5.1 Parámetros Khalil – Kleinfinger

La matriz de transformación está expresada como función de cuatro parámetros geométricos:

- α_j : el ángulo entre Z_{j-1} y Z_j alrededor de X_{j-1} .
- d_j : la distancia entre Z_{j-1} y Z_j a lo largo de X_{j-1} .
- θ_j : el ángulo entre X_{j-1} y X_j alrededor de Z_j .
- r_j : la distancia entre X_{j-1} y X_j a lo largo de Z_j .

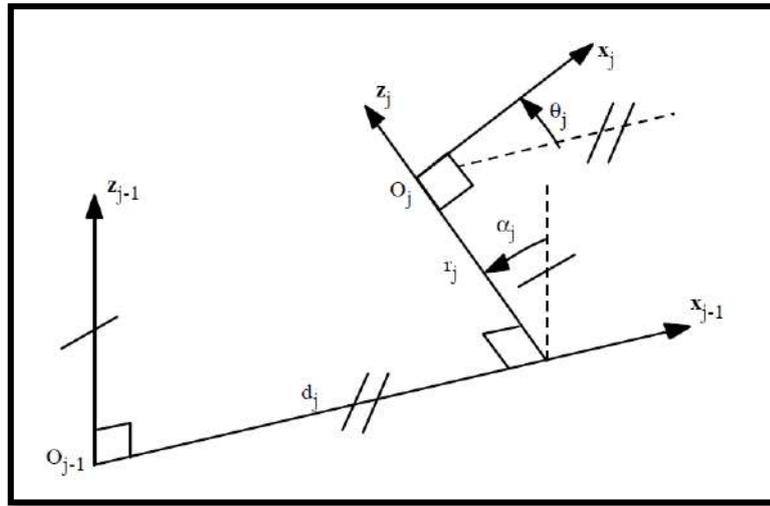


Figura 3.4 Parámetros geométricos Khalil – Kleinfinger ^[28]

Por lo tanto la matriz que define a un sistema j respecto a $j-1$ está dado por

$${}^{j-1}T_j = Rot(x, \alpha_j) * Trans(x, d_j) * Rot(z, \theta_j) * Trans(z, r_j) \quad (3.4)$$

$${}^{j-1}T_j = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\alpha_j * S\theta_j & C\alpha_j * C\theta_j & -S\alpha_j & -r_j * S\alpha_j \\ S\alpha_j * S\theta_j & S\alpha_j * C\theta_j & C\alpha_j & r_j * C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Mientras que la matriz que define un sistema $j-1$ respecto a j está dado por

$${}^jT_{j-1} = Trans(z, -r_j) * Rot(z, -\theta_j) * Trans(x, -d_j) * Rot(x, -\alpha_j) \quad (3.6)$$

$${}^jT_{j-1} = \begin{bmatrix} C\theta_j & C\alpha_j * S\theta_j & S\alpha_j * S\theta_j & -d_j * C\theta_j \\ -S\theta_j & C\alpha_j * C\theta_j & S\alpha_j * C\theta_j & d_j * S\theta_j \\ 0 & -S\alpha_j & C\alpha_j & -r_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.7)$$

3.5.2 Parámetros Denavit - Hartenberg

Los cuatro parámetros geométricos son:

- α_i : el ángulo entre Z_{i-1} y Z_i alrededor de X_i .
- a_i : la distancia entre Z_{i-1} y Z_i a lo largo de X_i .
- θ_i : el ángulo entre X_{i-1} y X_i alrededor de Z_{i-1} .
- d_i : la distancia entre X_{i-1} y X_i a lo largo de Z_{i-1} .

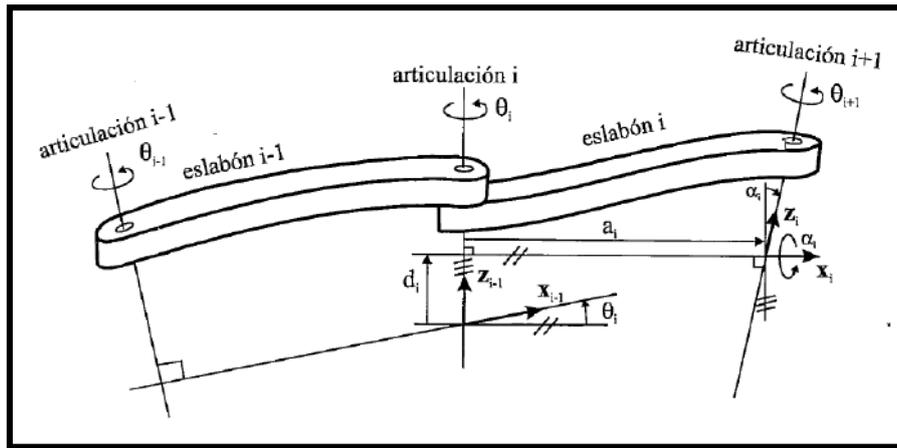


Figura 3.5 Parámetros geométricos Denavit - Hartenberg ^[31]

Por lo tanto la matriz que define a un sistema $\{S_i\}$ respecto a $\{S_{i-1}\}$ está dado por

$${}^{i-1}T_i = Rot(z, \theta_i) * Trans(z, d_i) * Trans(x, a_i) * Rot(x, \alpha_i) \quad (3.8)$$

$${}^{i-1}T_i = \begin{bmatrix} C\theta_i & -C\alpha_i * S\theta_i & S\alpha_i * S\theta_i & a_i * C\theta_i \\ S\theta_i & C\alpha_i * C\theta_i & -S\alpha_i * C\theta_i & a_i * S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

Mientras que su matriz inversa, es decir, la que define un sistema $\{S_{i-1}\}$ respecto a $\{S_i\}$ está dada:

$${}^i T_{i-1} = \begin{bmatrix} C\theta_i & S\theta_i & 0 & -a_i \\ -C\alpha_i * S\theta_i & C\alpha_i * C\theta_i & S\alpha_i & -d_i * S\alpha_i \\ S\alpha_i * S\theta_i & -S\alpha_i * C\theta_i & C\alpha_i & -d_i * C\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

3.6 ALGORITMO DE DENAVIT HARTENBERG

Para que la matriz ${}^i T_j$ relacione correctamente los sistemas $\{S_{i-1}\}$ respecto a $\{S_i\}$ es necesario que los sistemas se hayan escogido de acuerdo a unas normas determinadas, que junto a los 4 parámetros Denavit Hartenberg nos permitirán la resolución del modelo geométrico directo e inverso.

DH 1. Numeramos los eslabones. Siendo 0 la base fija. ($link = 0, 1, \dots, n$)

DH 2. Numeramos las articulaciones. ($joint = 1, \dots, n$)

DH 3. Reconocemos los ejes de cada articulación.

DH 4. Situamos el eje z_i del eslabón i en la articulación $i+1$ ($i = 1, \dots, n - 1$).

DH 5. Situamos el origen $\{S_0\}$ en cualquier punto del eje z_0 . Se asigna también x_0

DH 6. Para los eslabones $(1, \dots, n - 1)$ situamos $\{S_i\}$, en la intersección de z_i con la normal común a $z_{i-1} \wedge z_i$.

Si ambos ejes $(z_{i-1} \wedge z_i)$ se intersectan, se coloca $\{S_i\}$ en el punto de corte.

Si ambos ejes $(z_{i-1} \wedge z_i)$ son paralelos, se coloca $\{S_i\}$ en la articulación $i+1$.

DH 7. Situar x_i de tal forma que $x_i \perp z_{i-1}$.

DH 8. Situar y_i de modo que forme un sistema dextrógiro con x_i y z_i .

DH 9. Situar $\{S_n\}$ en el extremo del robot de modo que sea paralelo a $\{S_{n-1}\}$.

Una vez establecidos los sistemas de coordenadas de acuerdo a los pasos anteriores, podemos determinar los parámetros DH ($\alpha_i, a_i, \theta_i, d_i$) vistos en la sección anterior.

3.7 MODELAMIENTO ^[28]

Para el diseño y control de un robot se requieren ciertos modelos matemáticos como:

- **Modelos de Transformaciones**, que expresa la relación entre el espacio operacional (en el que la posición del efector final está definida) y el espacio de la articulación (en el cual, la configuración del robot está definida).

Estas transformaciones son importantes ya que los robots son controlados en el espacio articular, mientras que las tareas son definidas en el espacio operacional. Se considera:

- Modelo geométrico directo (MGD) e inverso (MGI).
 - Modelo cinemático directo e inverso.
- **Modelos dinámicos**, que da las relaciones entre las fuerzas o torques de los actuadores y las posiciones, velocidades y aceleraciones de las articulaciones.

3.7.1 Modelo geométrico directo (MGD):

Que da la localización y orientación del efector final como función de las coordenadas de las articulaciones. Para que el modelamiento de robots sea de una forma sistemática, se requiere de un método adecuado que permita describir su estructura.

Varios métodos y notaciones se han propuesto, siendo el más popular Denavit – Hartenberg (figura 2.4). Este método se desarrolló para estructuras seriales y presenta ciertas ambigüedades cuando se aplica a robots con cadenas cerradas. Por esta razón, en algunos trabajos se usa la notación de Khalil – Kleinfinger^[30], la cual da una descripción única para todos los sistemas articulados con un mínimo número de parámetros.

En el presente trabajo, sin embargo, se utilizarán ambas convenciones, que han sido descritos en el apartado anterior. Para una cadena cinemática serial se puede representar por la matriz 0T_n :

$${}^0T_n = {}^0T_1(q_1) {}^1T_2(q_2) {}^2T_3(q_3) \dots {}^{n-1}T_n(q_n) \quad (3.11)$$

Donde $\bar{q} = q_1, q_2, q_3, \dots, q_n$ son las variables articulares.

3.7.2 Modelo geométrico inverso (MGI)

Se debe se debe determinar las variables de las articulaciones a partir de una posición y orientación del efector final. Sin embargo, su cálculo es más complejo ya que las ecuaciones por resolver, por lo general, son no lineales, pueden existir varias soluciones o puede no existir ninguna debido a la arquitectura del robot.

$$\begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \end{bmatrix} \rightarrow \boxed{\text{CINEMATICA_INVERSA}} \rightarrow q_1, q_2, q_3, \dots, q_n$$

Para su cálculo, es habitual emplear *métodos geométricos*, que consisten en el uso de relaciones trigonométricas que permitan encontrar una solución de la cadena cinemática. Algunas ecuaciones usadas en el MGI se muestran en la tabla 2.2.

| | <i>Ecuación</i> | <i>Solución</i> |
|---|--|---|
| 1 | $a * \sin \theta + b * \cos \theta = c$ | $\theta = \text{atan2}(a, b) \mp \text{atan2}(\sqrt{a^2 + b^2 - c^2}, c)$ |
| 2 | $a * \sin \theta + b * \cos \theta = 0$ | $\theta = \text{atan2}(-b, a)$ ó $\theta = \text{atan2}(b, -a)$ |
| 3 | $\cos \theta = a \wedge \sin \theta = b$ | $\theta = \text{atan2}(b, a)$ |
| 4 | $\cos \theta = a$ | $\theta = \text{atan2}(\mp \sqrt{1 - a^2}, a)$ |
| 5 | $\sin \theta = a$ | $\theta = \text{atan2}(a, \mp \sqrt{1 - a^2})$ |

Tabla 3.1 Ecuaciones trigonométricas usadas en cinemática inversa.

También se puede utilizar las *matrices de transformación homogénea* mediante el método de Paul, en el cual se pre-multiplica por las matrices inversas (ecuaciones 3.6) iterativamente, de forma que se pueda aislar una a una las variables en función de las demás.

Además, es posible utilizar el principio de *desacoplo cinemático* en robots en los que sus tres últimos grados de libertad situados al final de la cadena cinemática se cortan en un punto, denominado muñeca del robot, y son utilizados para la orientación del efector final. En este caso se calcula primero las variables encargadas de posicionar la muñeca y posteriormente el resto.

Por otro lado, se puede emplear también *métodos numéricos* que pueden ser programados por computador, sin embargo el principal inconveniente es la velocidad de convergencia, ya que algunas aplicaciones deben resolverse en tiempo real, e incluso su convergencia en sí no está asegurada. Entre estos métodos tenemos el método de la bisección y el método de newton raphson.

3.7.2.1 Métodos Numéricos: Método de la Bisección

En el desarrollo del MGI existen también procedimientos iterativos que pueden programarse para el cálculo de alguna de las variables. Uno de ellos es el método de la bisección.

El *Método de la Bisección* se basa en dos teoremas:

- Teorema de Bolzano
- Teorema del Valor Intermedio.

Teorema de Bolzano: si una función es continua en un intervalo cerrado y acotado, y en las cotas el valor de las funciones son de signos opuestos, entonces existe al menos una raíz de la función dentro del intervalo. Sea $f: [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$ una función continua en $[a, b]$ tal que $f(a) \cdot f(b) < 0$, entonces, existe un $c \in (a, b)$ tal que $f(c) = 0$.

Teorema del Valor Medio: Sea $f: [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$ continua en $[a, b]$, tal que $f(a) < f(b)$ entonces, $\forall k$ tal que $f(a) < k < f(b)$ existe un $x_0 \in (a, b)$ tal que $f(x_0) = k$.

El método de la bisección se utiliza para aproximar la solución de una ecuación que es igual a cero. Si deseamos encontrar los ceros de la función $f(x)$ continua, debemos tener dos puntos a y b tal que $f(a)$ y $f(b)$ tengan signos distintos, y por el teorema de Bolzano $f(x)$ debe existir, al menos, una raíz en ese intervalo $[a, b]$.

El método usa un tercer punto $c = \frac{a+b}{2}$, que es la semisuma del intervalo anterior y genera dos nuevos intervalos $[a, c]$ y $[c, b]$. Se evalúa en cuál de los dos sus funciones tienen signo diferente, es decir, $f(a)$ y $f(c)$, ó $f(c)$ y $f(b)$ y se aplica nuevamente el método a dicho intervalo. Este procedimiento se repite hasta alcanzar la tolerancia requerida.

3.7.3 Modelo Cinemático Directo e Inverso ^[31]

El modelamiento cinemático de un robot busca conocer la relación entre las velocidades de las coordenadas articulares y las velocidades de la posición y orientación del extremo del laparoscopio, es decir, en qué afecta un movimiento diferencial de las articulaciones al movimiento final en el espacio de la tarea.

El modelo diferencial es determinado por la matriz Jacobiana, la cual permite relacionar el vector de velocidades articulares $(\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n)$ con otro vector de velocidades expresado en un espacio distinto. Podemos distinguir la Jacobiana analítica y la Jacobiana geométrica.

En ambos casos, la Jacobina Directa permite conocer las velocidades del extremo del robot a partir de los valores de cada articulación, mientras que la Jacobiana Inversa permitirá conocer las velocidades articulares necesarias para obtener un vector concreto de velocidades del extremo.

3.7.3.1 Jacobiana Analítica

Que relaciona el vector de variables articulares (q_1, q_2, \dots, q_n) con el vector de velocidades de la posición y orientación del extremo del robot, expresado en sus ángulos de Euler $(\dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi})$.

Sean:

$$\begin{aligned} x &= f_x(q_1, q_2, \dots, q_n), \quad y = f_y(q_1, q_2, \dots, q_n), \quad z = f_z(q_1, q_2, \dots, q_n) \\ \phi &= f_\phi(q_1, q_2, \dots, q_n), \quad \theta = f_\theta(q_1, q_2, \dots, q_n), \quad \psi = f_\psi(q_1, q_2, \dots, q_n) \end{aligned}$$

Si derivamos con respecto al tiempo:

$$\begin{aligned} \dot{x} &= \sum_1^n \frac{\partial f_x}{\partial q_i} \dot{q}_i, & \dot{y} &= \sum_1^n \frac{\partial f_y}{\partial q_i} \dot{q}_i, & \dot{z} &= \sum_1^n \frac{\partial f_z}{\partial q_i} \dot{q}_i \\ \dot{\phi} &= \sum_1^n \frac{\partial f_\phi}{\partial q_i} \dot{q}_i, & \dot{\theta} &= \sum_1^n \frac{\partial f_\theta}{\partial q_i} \dot{q}_i, & \dot{\psi} &= \sum_1^n \frac{\partial f_\psi}{\partial q_i} \dot{q}_i \end{aligned}$$

Expresado en forma matricial:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = J_a \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad (3.12)$$

Donde $J_a = \begin{bmatrix} \frac{\partial f_x}{\partial q_1} & \dots & \frac{\partial f_x}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_\psi}{\partial q_1} & \dots & \frac{\partial f_\psi}{\partial q_n} \end{bmatrix}$ se denomina matriz Jacobiana analítica.

3.7.3.2 Jacobiana Geométrica

Relaciona el vector de variables articulares $(\dot{q}_1, \dot{q}_2, \dots, \dot{q}_n)$ con las velocidades lineales (\mathbf{v}) y velocidades angulares (\mathbf{w}) del extremo del robot $(v_x, v_y, v_z, w_x, w_y, w_z)$. Sea:

$$\begin{bmatrix} v_x \\ v_y \\ v_z \\ w_x \\ w_y \\ w_z \end{bmatrix} = \mathbf{J} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \vdots \\ \dot{q}_n \end{bmatrix} \quad (3.13)$$

- La velocidad lineal del extremo del robot las cuales están expresadas en el sistema de referencia de la base del robot $\{S_0\}$, será:

$$v_x = \frac{dx}{dt} = \dot{p}_x, \quad v_y = \frac{dy}{dt} = \dot{p}_y, \quad v_z = \frac{dz}{dt} = \dot{p}_z \quad (3.14)$$

- La relación de la velocidad angular se determinará a partir de la submatriz (3x3) de rotación $\mathbf{R} = [\vec{n} \ \vec{o} \ \vec{a}]$ obtenida de la matriz de transformación homogénea del robot \mathbf{T} . La matriz \mathbf{R} es ortonormal, por lo que $\mathbf{R} \cdot \mathbf{R}^T = \mathbf{I}$.

Si derivamos con respecto al tiempo tendremos $\dot{\mathbf{R}} \cdot \mathbf{R}^T + \mathbf{R} \cdot \dot{\mathbf{R}}^T = \mathbf{0}$.

Ahora definimos la matriz $\boldsymbol{\Omega} = \dot{\mathbf{R}} \cdot \mathbf{R}^T$. Se ha demostrado ^[32] que esta matriz está relacionada con las velocidades angulares, según:

$$\boldsymbol{\Omega} = \begin{bmatrix} \mathbf{0} & -w_z & w_y \\ w_z & \mathbf{0} & -w_x \\ -w_y & w_x & \mathbf{0} \end{bmatrix} \quad (3.15)$$

Por lo tanto, con las ecuaciones 3.10 y 3.11 es posible calcular todos los términos de la Jacobiana \mathbf{J} .

Sin embargo, muchas veces se necesita el valor instantáneo de la Jacobiana, es decir su valor numérico. El siguiente procedimiento está basado en la *propagación de velocidades*. Este método permite obtener las columnas de la matriz Jacobiana geométrica medida con respecto al sistema base $\{S_0\}$, a partir de las matrices ${}^{i-1}T_i$.

$${}^{i-1}T_i = \begin{bmatrix} {}^{i-1}\vec{n}_i & {}^{i-1}\vec{o}_i & {}^{i-1}\vec{a}_i & {}^{i-1}\vec{P}_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

Si las matrices de transformación homogénea son definidas como en 3.12 y contienen la información de la orientación y posición del sistema $\{S_i\}$ con respecto a $\{S_{i-1}\}$. Entonces, la matriz 0T_i contendrá los vectores directores y origen del sistema $\{S_i\}$ con respecto al sistema base $\{S_0\}$. Por lo tanto, se define ${}^0\mathbf{z}_i$ como:

$${}^0\mathbf{z}_i = {}^0T_i(1:3,3) \quad (3.17)$$

El vector que va del origen del sistema $\{S_i\}$ hasta el extremo del robot, es decir el sistema $\{S_n\}$, será denominado ${}^i\mathbf{p}_n$ y se obtendrá como:

$${}^i\mathbf{p}_n = {}^0\mathbf{T}_n(1:3,4) - {}^0\mathbf{T}_i(1:3,4) \quad (3.18)$$

Entonces la matriz Jacobiana será una matriz de $(6 \times n)$, donde $n = \text{grados de libertad}$ y quedará expresada como:

$$J = [J_1 | J_2 \dots | J_n] \quad (3.19)$$

Donde:

$$J_i = \begin{cases} \begin{bmatrix} {}^0\mathbf{z}_{i-1}^X {}^{i-1}\mathbf{p}_n \\ {}^0\mathbf{z}_{i-1} \end{bmatrix}, & \text{Si el eslabón } i \text{ es de rotación} \\ \begin{bmatrix} {}^0\mathbf{z}_{i-1} \\ \vec{0} \end{bmatrix}, & \text{Si el eslabón } i \text{ es de traslación} \end{cases}$$

3.7.4 Modelo Dinámico Directo e Inverso ^[31]

El modelamiento dinámico inverso permite relacionar las posiciones, velocidades y aceleraciones angulares con los momentos o pares necesarios para producir dicho movimiento. Se puede resolver analíticamente mediante las formulaciones Lagrangiana y el método de Newton-Euler. Además se puede utilizar el entorno de Matlab, en la cual existen librerías ya creadas para su cálculo. Por otro lado, se puede hacer uso de diseños 3D e importarlos al entorno de Matlab mediante SimMechanics y también obtener los torques en las articulaciones. El modelo dinámico directo permite la relación inversa, es decir, obtener las posiciones y velocidades articulares a partir de los pares ejercidos.

3.8 CONTROL ^[27]

En el control de un robot, las articulaciones se accionan mediante actuadores. Éstos aplican una fuerza o un par de torsión sobre los eslabones para su movimiento. Sus comandos provienen del sistema de control del manipulador que lo mueven para ejecutar el movimiento específico requerido. A continuación se presenta un diagrama esquemático del control de un manipulador.

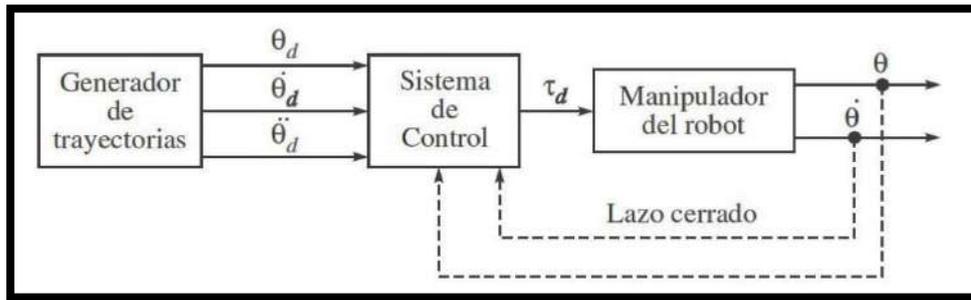


Figura 3.6 Diagrama de bloques de un sistema de control de robots ^[27]

Para el control de robots existen diversas técnicas como el control cinemático, adaptativo, el par de torsión computado, el difuso, entre otros. Además, debido a la naturaleza no lineal de los manipuladores se han estudiado otros métodos como el control lineal, en el que se trata de forma independiente a cada articulación. A éste se le puede aplicar controles clásicos como el PID que son bastante más robustos en este tipo de tareas.

3.8.1 Motor DC

Las articulaciones se accionan generalmente por motores de corriente continua. Un motor de corriente continua es una máquina que convierte la energía eléctrica en mecánica, provocando un movimiento rotatorio.

Está compuesto básicamente de dos partes: El estator (también llamado campo) que da soporte mecánico al aparato y contiene los devanados principales de la máquina. Encargado de crear el campo magnético, que puede ser a través de imanes permanentes o electroimanes. El rotor (también llamado armadura) está formado por un núcleo ferromagnético con devanados, que es en donde se induce la fuerza electromotriz. Es alimentado con corriente directa generalmente por escobillas.

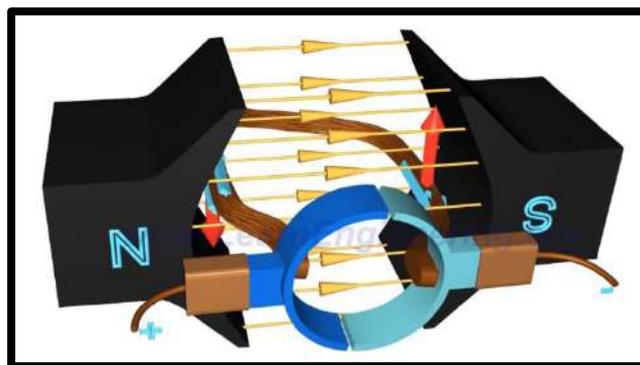


Figura 3.7 Principio de funcionamiento motor DC (Fuente: <https://www.youtube.com/watch?v=aqUJi2WsGjk>)

Su principio de funcionamiento se muestra en la Figura 3.7. Está basado en la ecuación de Lorentz, que establece que en presencia de un campo magnético (flechas amarillas), si por un conductor circula una corriente (flechas celestes), en éste se ejercerá una fuerza perpendicular a ambas (flechas rojas).

Hay cinco tipos de motores dc de uso general ^[40]:

- Motor dc de excitación separada
- Motor dc en derivación
- Motor dc de imán permanente
- Motor dc en serie
- Motor dc compuesto.

Actualmente existen otros tipos motores DC ^[41], en los cuales el bobinado de cobre no presenta núcleo de hierro. Éstos presentan ventajas comparativas como una alta eficiencia, momento de inercia muy bajo, baja inductancia, bajas emisiones electromagnéticas y compactos. Dentro de ellos, hay los tradicionalmente con escobillas y los llamados BLDC (Brushless DC motor). En el apartado 5.1 se explica con más detalle sus ventajas.

3.9 SIMULACIÓN

Existen diversos software que permiten la etapa de simulación. Inicialmente es necesario crear un modelo en 3D que permita representar el brazo robótico físico. Para esto se ha decidido utilizar SolidWorks. Posteriormente, para simular el comportamiento y el seguimiento de una determinada trayectoria, así como el uso de sensores y actuadores, se ha utilizado el entorno de simulación SimMechanics dentro de Matlab.

3.9.1 SolidWorks

SolidWorks es un software de diseño 3D que permite modelar piezas y ensamblajes. Permite la asignación de materiales y cambiar la apariencia de las mismas. A partir de ellos se pueden crear planos técnicos para su documentación. Además opcionalmente permite realizar análisis de esfuerzos, estudio del movimiento y análisis de fluidos.

3.9.2 SimMechanics ^[42]

SimMechanics (ahora llamado Simscape Multibody) proporciona un entorno de simulación para sistemas mecánicos 3D como robots, suspensión en vehículos, equipos de construcción, etc. Permite modelar sistemas multi-cuerpo usando bloques que representan los cuerpos, articulaciones, restricciones, elementos de fuerza y sensores. SimMechanics formula y resuelve las ecuaciones de movimiento para completar el sistema mecánico. Además permite importar ensamblajes completos CAD, incluyendo las masas, inercias, articulaciones y toda la geometría en 3D a tu modelo, generando una animación 3D que permite visualizar el sistema dinámico.

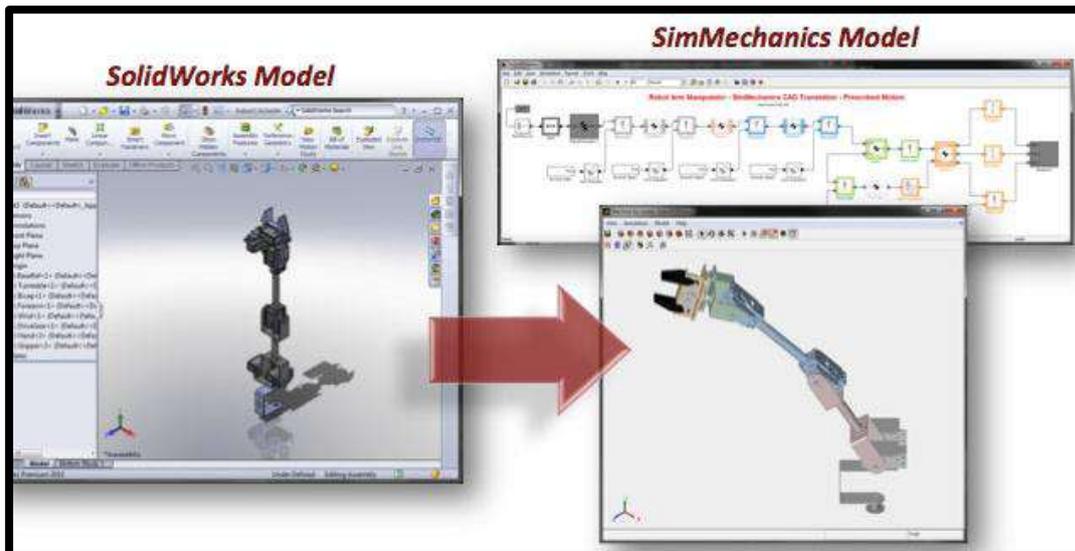


Figura 3.8 Diseño mecánico en Solidworks y SimMechanics.

(Fuente: <https://es.mathworks.com/content/dam/mathworks/mathworks-dot-com/images/nextgen/thumbnails/events/matlab-tour-cn-2015.jpg>)

Capítulo 4

Diseño y Modelamiento del Asistente Robótico

Para definir la estructura de nuestro robot (lo llamaremos IMEbot), se ha tomado en cuenta el análisis de las estructuras funcionales de robots aplicados anteriormente a cirugía mínimamente invasiva (Apartado 1.2.3). Además, para realizar el modelamiento matemático, debemos tomar en cuenta la restricción del paso por el trocar, mientras que el diseño en 3D de los cuerpos (eslabones) nos ayudará a poder extraer los valores numéricos de los parámetros dinámicos.

4.1 DISEÑO IMEBOT

Una vez analizadas las estructuras funcionales de los principales robots utilizados en cirugía, procedemos a definir la estructura que tendrá nuestro asistente robótico para la manipulación de la cámara dentro de la cirugía laparoscópica. Este deberá cumplir lo siguiente:

1. Tener un grado de libertad que permita ajustar a qué altura se debe posicionar el robot, debido a que la mayoría de camillas son regulables.
2. Poder posicionarse correctamente dentro del volumen de trabajo en el abdomen del paciente.
3. Tener un grado de libertad que permita el giro del laparoscopio sobre su eje, permitiendo utilizar otros tipos (0° , 30° , 45° y 70°) de laparoscopio y no sólo el de 0° , ya que para éstos es necesario conocer tanto su posición como la orientación.
4. Tomar en cuenta que el paso del laparoscopio por el trocar debe ser un punto de pivote fijo, ya que no debería existir un sobreesfuerzo en esa parte del abdomen, evitando que la incisión se acreciente.
5. Conocer la posición final, fijando el último sistema de coordenadas en el extremo distal del laparoscopio y que servirá como referencia para el modelamiento geométrico.

Por lo tanto, IMEbot tendrá las siguientes características:

- **Articulación 1:**
Es una articulación prismática que permitirá regular la altura a la que se posicionará el robot, esto dependerá de la altura de la camilla.
- **Articulaciones 2, 3 y 4:**
Son tres articulaciones rotacionales activas que permitan posicionar el laparoscopio de acuerdo a los requerimientos del médico.
- **Articulación 5 y 6:**
Son dos articulaciones rotacionales que permiten mantener el paso del trócar como pivote, ya que es un punto fijo.
- **Articulación 7:**
Una articulación rotacional activa que permite girar el laparoscopio sobre su eje, que servirá cuando se use laparoscopios con diferentes ángulos de visión. Éste es actuado independientemente de acuerdo a los requerimientos del médico

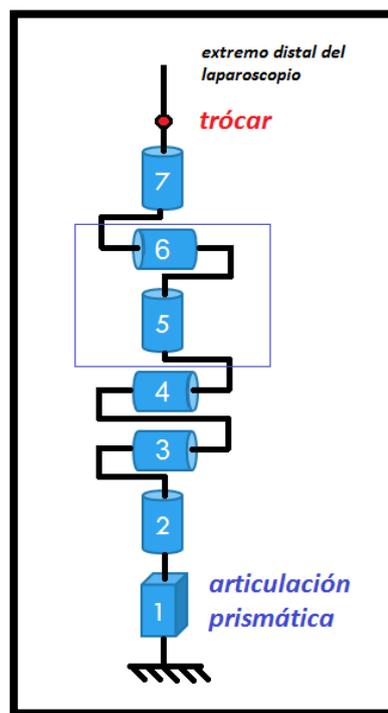


Figura 4.1 Estructura cinemática IMEbot (elaboración propia)

Resumiendo, se optó para el diseño de IMEbot una estructura funcional de 7 grados de libertad, con 1 articulación prismática regulable, 6 articulaciones rotacionales activas (motorizadas) que permitirán posicionar y orientar sobre su eje el laparoscopio de acuerdo a lo requerido. Además, éstas servirán para mantener fijo el movimiento en el trócar y el giro del laparoscopio sobre sus otros dos ejes coordenados. En la Figura 4.1 se puede observar la estructura cinemática propuesta.

4.2 DISEÑO 3D IMEBOT

Para el dimensionamiento de IMEbot se ha tomado en cuenta las siguientes consideraciones:

- Las camillas empleadas en las salas de operaciones de la clínica San Miguel (una de las más importantes en la ciudad de Piura) son las mesas quirúrgicas marca AMTAI modelo T800LSK, cuyas medidas son: - Altura mínima = 0.68 m. – Ancho = 0.58 m. [34]. Éstas han sido tomado como referencia para las medidas de los eslabones y dado que la camilla es regulable, consideraremos una altura total de 0.70 m.
- Se ha considerado una altura promedio de 30 cm, medidos desde la parte superior de la camilla, que representa la altura del abdomen sumado a su elevación que se produce durante la insuflación con dióxido de carbono.
- El material asignado en Solidworks para la base y los eslabones 1, 2, 3, 4, 5 y 6 es *Aleación de aluminio 6062 – O (SS)* cuya densidad es de 2700 kg/m³. Para el laparoscopio se utilizó *Acero AISI 316 Barra de acero inoxidable recocido (SS)* cuya densidad es 8000 kg/m³.
- Debido a la longitud del trócar que va al interior del abdomen, el extremo distal del laparoscopio deberá estar insertado por lo menos 5 cm dentro del paciente.

Teniendo en cuenta lo anterior, las dimensiones del asistente robótico IMEbot se adjuntan en los planos mostrados en la parte final del presente documento, los cuales, son de mucha importancia ya que es la propuesta real de la implementación de un posible prototipo.

En la figura 4.2 se muestran las piezas y/o ensambles de la base y los cuerpos de los eslabones 1 y 2 realizados en el software SOLIDWORKS.

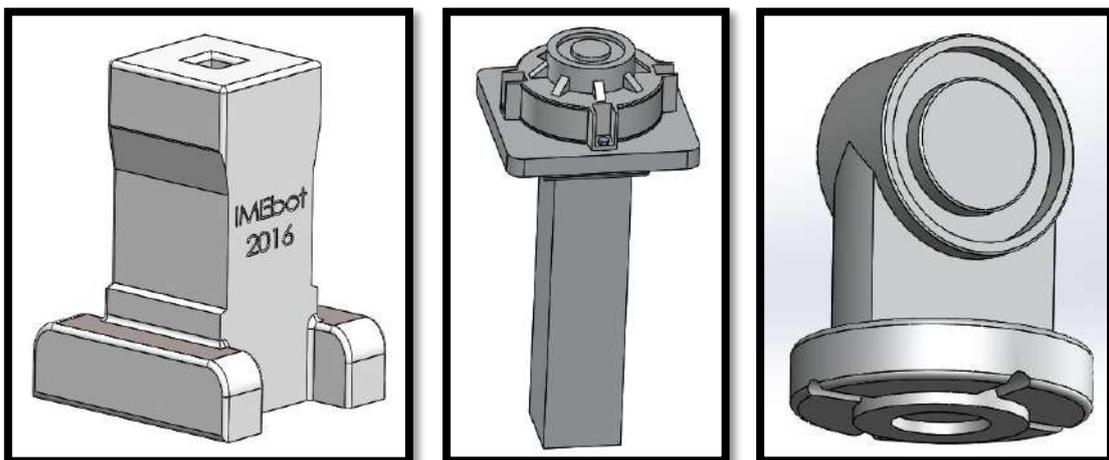


Figura 4.2 Izquierda: Base; Centro: Eslabón 1; Derecha: Eslabón 2

En la figura 3.3 se muestran los cuerpos de los eslabones 3 y 4.

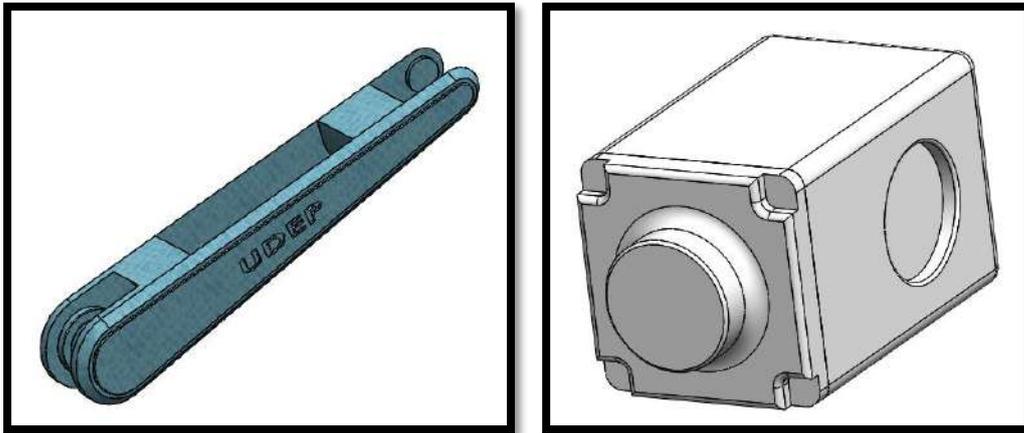


Figura 4.3 Izquierda: Eslabón 3; Derecha: Eslabón 4

En la figura 4.4 se muestra el cuerpo de los eslabones 5 y 6:

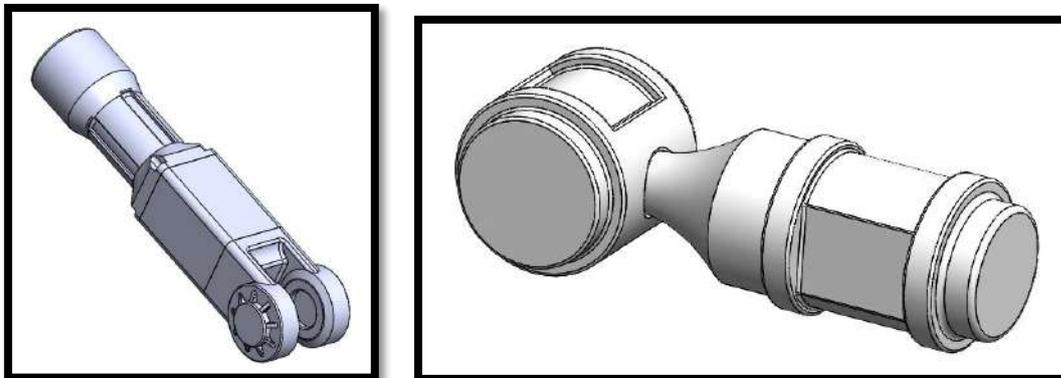


Figura 4.4 Izquierda: Eslabón 5; Derecha: Eslabón 6

En la figura 4.5 se muestra la articulaciones 7, que sería el laparoscopio:

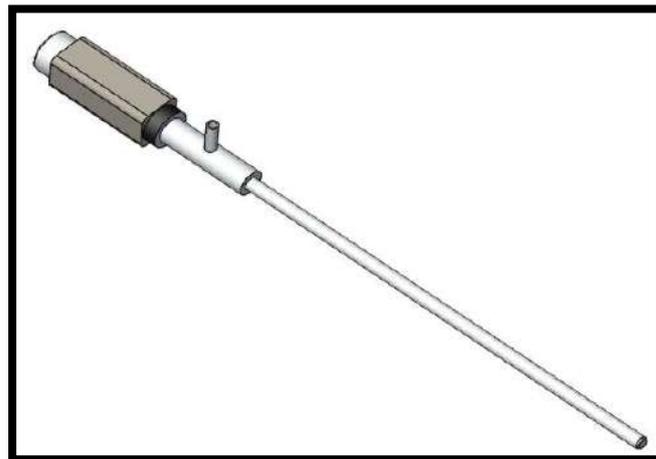


Figura 4.5 Eslabón 7 (laparoscopio)

Cabe mencionar que las articulaciones 2, 3 y 4 son las responsables de la posición final del laparoscopio, mientras las articulaciones 5, 6 y 7 son las encargadas de su orientación final. Además, si se usa un laparoscopio de 0° , es decir, la imagen proyectada es exactamente

perpendicular a la enfocada por el laparoscopio, no es necesario variar el ángulo de la articulación 7. El ensamble total se muestra en la figura 4.6:

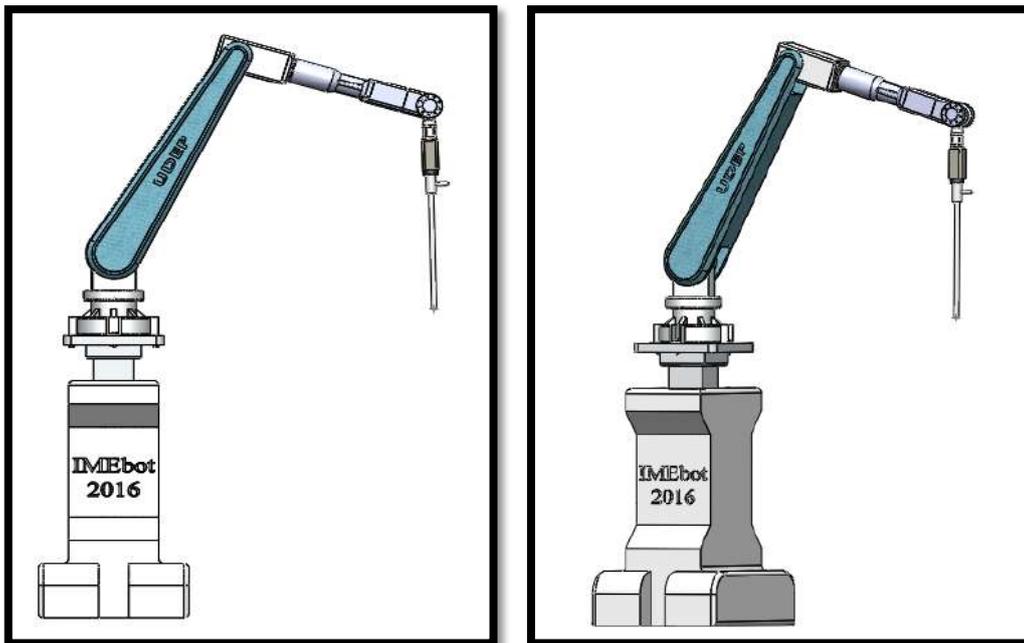


Figura 4.6 Ensamble final (asistente robótico)

4.3 MODELAMIENTO GEOMÉTRICO IMEBOT

Una vez definida la estructura del robot y realizado el diseño de las piezas mecánicas en 3D, procederemos a calcular los modelos matemáticos, a fin de que IMEbot pueda realizar los movimientos requeridos por el laparoscopio

Se debe tener en cuenta el paso del trócar que se mantendrá como un punto de pivote fijo, con el fin de evitar el agrandamiento de la incisión y que la recuperación del paciente sea satisfactoria.

Para obtener estos modelos geométricos, se ha abordado el tema de con dos métodos distintos:

- Primero, utilizando los parámetros Khalil – Kleinfinger ^[24] para el cálculo del modelo geométrico directo y el método de Paul para su modelo geométrico inverso. En este último también se aplicará métodos numéricos para hallar uno de los ángulos.
- Segundo, utilizando los parámetros Denavit – Hartenberg (que es la convención más usada) para el cálculo del modelo geométrico directo y mediante desacoplo cinemático se calcula el modelo geométrico inverso.

Si bien ambos cumplen la misma función y se obtienen resultados similares, se ha creído conveniente ahondar en los mismos. Además, es importante tener en cuenta que el cálculo de los algoritmos debe ser en tiempo real, siendo necesario analizar la carga computacional que se emplea para desarrollar ambos algoritmos, sobretodo haciendo énfasis en el modelo geométrico inverso, y por ende se debe elegir la mejor opción.

4.3.1 Modelamiento Geométrico Directo (MGD)

El modelamiento geométrico directo (MGD) nos permite conocer la posición y orientación final del laparoscopio en el espacio de trabajo (dentro del abdomen del paciente) a partir de los ángulos articulares y la posición de la base del robot.

Este modelo es muy importante ya que permite conocer la posición final del laparoscopio sin necesidad de un sistema de visión artificial.

4.3.1.1 Parámetros Khalil – Kleinfinger

Los parámetros obtenidos son:

| j | σ_j | M | α_j | d_j | θ_j | r_j |
|-----|------------|-----|------------|-------|------------------------------|-------------|
| 1 | 1 | 1 | 0° | 0 | 0 | $r_1 = q_1$ |
| 2 | 0 | 1 | 0° | 0 | $90^\circ + \theta_2 = q_2$ | 0 |
| 3 | 0 | 1 | 90° | 0 | $\theta_3 = q_3$ | 0 |
| 4 | 0 | 1 | 0° | d_3 | $-90^\circ + \theta_4 = q_4$ | 0 |
| 5 | 0 | 1 | -90° | 0 | $\theta_5 = q_5$ | r_5 |
| 6 | 0 | 1 | 90° | 0 | $90^\circ + \theta_6 = q_6$ | 0 |
| 7 | 0 | 1 | 90° | 0 | $\theta_7 = q_7$ | r_7 |

Tabla 4.1 Parámetros geométricos Khalil – Kleinfinger

Donde:

- j : Número de la articulación.
- σ_j : Indica si la articulación es revoluta ($\sigma = 0$) o prismática ($\sigma = 1$).
- M : Indica si la articulación es pasiva ($M = 0$) o activa ($M = 1$).
- α_j : el ángulo entre Z_{j-1} y Z_j alrededor de X_{j-1} .
- d_j : la distancia entre Z_{j-1} y Z_j a lo largo de X_{j-1} .
- θ_j : el ángulo entre X_{j-1} y X_j alrededor de Z_j .
- r_j : la distancia entre X_{j-1} y X_j a lo largo de Z_j .

Las dimensiones de las piezas diseñadas anteriormente, esto es, medidas desde el origen de los sistemas coordenados asignados, se representan simbólicamente en las variables:

- $r_1 = \text{base} + \text{eslabón 1} + \text{eslabón 2} = 1.05m$
- $d_3 = \text{eslabón 3} = 0.5m$
- $r_5 = \text{eslabón 4} + \text{eslabón 5} = 0.4m$
- $r_7 = \text{eslabón 6} + \text{eslabón 7 (laparoscopio)} = 0.45m$

En la figura a continuación se muestra los valores numéricos de las variables mencionadas y la ubicación de los sistemas de coordenadas asignados para cada eslabón. Estos permiten encontrar los parámetros de cada sistema de coordenadas que se muestran en la tabla 4.1.

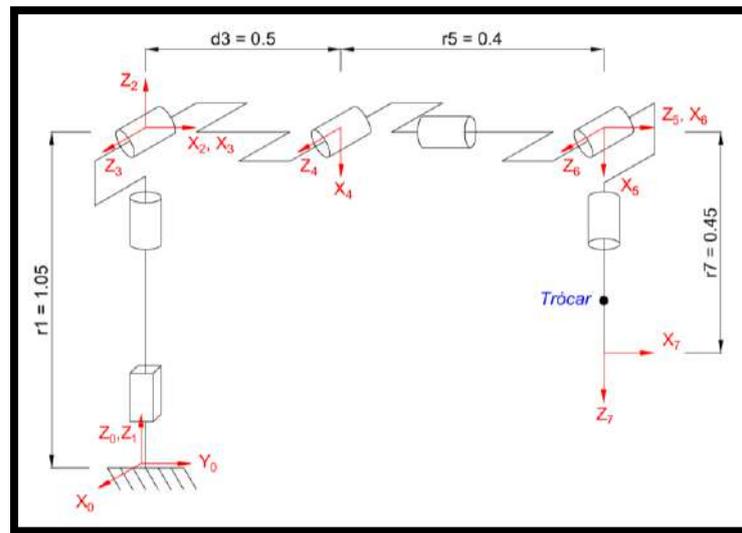


Figura 4.7 Sistema de coordenadas de los eslabones con parámetros Khalil - Kleinfinger
(elaboración propia)

Las matrices de transformación homogénea obtenidas para cada articulación se muestran en la ecuación 4.1.

$$\begin{aligned}
 {}^0T_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; & {}^1T_2 &= \begin{bmatrix} c_2 & -s_2 & 0 & 0 \\ s_2 & c_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \\
 {}^2T_3 &= \begin{bmatrix} c_3 & -s_3 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_3 & c_3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; & {}^3T_4 &= \begin{bmatrix} c_4 & -s_4 & 0 & d_3 \\ s_4 & c_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \\
 {}^4T_5 &= \begin{bmatrix} c_5 & -s_5 & 0 & 0 \\ 0 & 0 & 1 & r_5 \\ -s_5 & -c_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; & {}^5T_6 &= \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^6T_7 &= \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ 0 & 0 & -1 & -r_7 \\ s_7 & c_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{4.1}$$

La imagen es transmitida al monitor desde extremo distal del laparoscopio, es decir, la ubicación de interés estará en el sistema de coordenadas 7, por lo que el MGD será determinado por la matriz de transformación ${}^0T_7 = \begin{bmatrix} {}^0s_7 & {}^0n_7 & {}^0a_7 & {}^0P_7 \end{bmatrix}$, donde ${}^0s_7, {}^0n_7, {}^0a_7$ son los vectores columna que representan la orientación y 0P_7 es un vector columna que representa la posición. Las ecuaciones obtenidas para la posición ${}^0P_7(X,Y,Z)$ (función ‘MGD_Khalil.m’, anexo 9.2.1.1.1) son:

$$X = c2 * c3 * d3 - c2 * s34 * (r5 - c6 * r7) - r7 * s2 * s5 * s6 + c2 * c5 * c34 * r7 * s6 \quad (4.2)$$

$$Y = c3 * d3 * s2 + s2 * s34 * (r5 - c6 * r7) + c2 * r7 * s5 * s6 + c5 * c34 * r7 * s2 * s6 \quad (4.3)$$

$$Z = r1 + d3 * s3 + c34 * (r5 - c6 * r7) + c5 * r7 * s6 * s34 \quad (4.4)$$

Donde:

- $c_i = \cos(\theta_i)$
- $s_i = \sin(\theta_i)$
- $c_{ij} = \cos(\theta_i + \theta_j)$
- $s_{ij} = \sin(\theta_i + \theta_j)$

La matriz de orientación 0A_7 tiene la forma general: ${}^0A_7 = \begin{bmatrix} s_x & n_x & a_x \\ s_y & n_y & a_y \\ s_z & n_z & a_z \end{bmatrix}$ cuando usamos

los parámetros Khalil – Kleinfinger. Por lo tanto las ecuaciones que representan la orientación del laparoscopio después de factorizar y reemplazar identidades trigonométricas son:

$$s_x = s2 * (c5 * s7 - c6 * c7 * s5) + c2 * c34 * (s5 * s7 + c5 * c6 * c7) - c7 * s6 * s34 \quad (4.5)$$

$$s_y = c34 * s2 * (s5 * s7 + c5 * c6 * c7) - c2 * (c5 * s7 - c6 * c7 * s5) - c7 * s2 * s6 * s34 \quad (4.6)$$

$$s_z = s34 * (s5 * s7 + c5 * c6 * c7) + c7 * c34 * s6 \quad (4.7)$$

$$n_x = s2 * (c5 * c7 + c6 * s5 * s7) + c2 * c34 * (c7 * s5 - c5 * c6 * s7) + c2 * s6 * s7 * s34 \quad (4.8)$$

$$n_y = c34 * s2 * (c7 * s5 - c5 * c6 * s7) - c2 * (c5 * c7 + c6 * s5 * s7) + s2 * s6 * s7 * s34 \quad (4.9)$$

$$n_z = s34 * (c7 * s5 - c5 * c6 * s7) - c34 * s6 * s7 \quad (4.10)$$

$$a_x = c2 * c6 * s34 - s2 * s5 * s6 + c2 * c5 * c34 * s6 \quad (4.11)$$

$$a_y = c2 * s5 * s6 + c6 * s2 * s34 + c5 * c34 * s2 * s6 \quad (4.12)$$

$$a_z = c5 * s6 * s34 - c6 * c34 \quad (4.13)$$

4.3.1.2 Parámetros Denavit - Hartenberg

Los parámetros Denavit – Hartenberg obtenidos después del análisis son:

| i | a_i | α_i | d_i | θ_i |
|-----|-------|-------------|------------|------------------------------|
| 1 | 0 | 0° | $D1 = q_1$ | 0° |
| 2 | 0 | -90° | L2 | $\theta_2 = q_2$ |
| 3 | A3 | 0° | 0 | $-90^\circ + \theta_3 = q_3$ |
| 4 | 0 | 90° | 0 | $180^\circ + \theta_4 = q_4$ |
| 5 | 0 | 90° | L5 | $180^\circ + \theta_5 = q_5$ |
| 6 | 0 | 90° | 0 | $-90^\circ + \theta_6 = q_6$ |
| 7 | 0 | 0° | L7 | $\theta_7 = q_7$ |

Tabla 4.2 Parámetros geométricos Denavit - Hartenberg

Donde:

- α_i : el ángulo entre Z_{i-1} y Z_i alrededor de X_i .
- a_i : la distancia entre Z_{i-1} y Z_i a lo largo de X_i .
- θ_i : el ángulo entre X_{i-1} y X_i alrededor de Z_{i-1} .
- d_i : la distancia entre X_{i-1} y X_i a lo largo de Z_{i-1} .

Se utilizó las piezas diseñadas y el ensamble final en Solidworks para asignar los orígenes de los sistemas de coordenadas de acuerdo al algoritmo presentado en el epígrafe 3.6, como se aprecia a continuación.

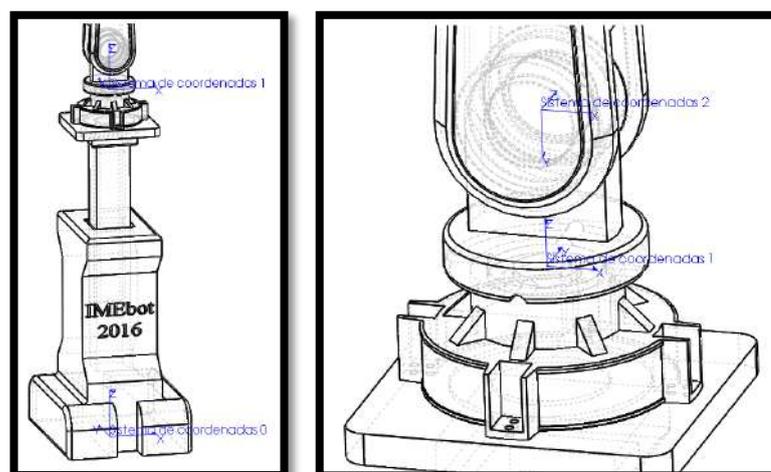


Figura 4.8 Sistema de coordenadas 1 y 2 con parámetros D-H (elaboración propia)

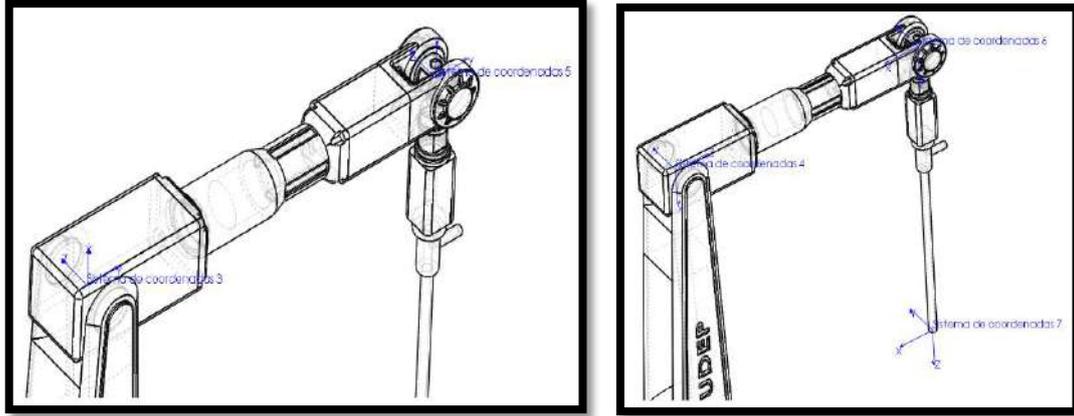


Figura 4.9 Sistema de coordenadas 3, 4, 5, 6 y 7 con parámetros D-H
(elaboración propia)

Las matrices de transformación homogénea obtenidas para cada articulación se muestran en la ecuación (4.14). Se hace uso de la función ‘MGD.m’ que se muestra en el anexo 9.2.1.2.1.

$$\begin{aligned}
 {}^0T_1 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & D1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^1T_2 = \begin{bmatrix} c_2 & 0 & -s_2 & 0 \\ s_2 & 0 & c_2 & 0 \\ 0 & -1 & 0 & L2 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \\
 {}^2T_3 &= \begin{bmatrix} s_3 & c_3 & 0 & A3*s_3 \\ -c_3 & s_3 & 0 & -A3*c_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^3T_4 = \begin{bmatrix} -c_4 & 0 & -s_4 & 0 \\ -s_4 & 0 & c_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \\
 {}^4T_5 &= \begin{bmatrix} -c_5 & 0 & -s_5 & 0 \\ -s_5 & 0 & c_5 & 0 \\ 0 & 1 & 0 & L5 \\ 0 & 0 & 0 & 1 \end{bmatrix}; {}^5T_6 = \begin{bmatrix} s_6 & 0 & -c_6 & 0 \\ -c_6 & 0 & -s_6 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
 {}^6T_7 &= \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ s_7 & c_7 & 0 & 0 \\ 0 & 0 & 1 & L7 \\ 0 & 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{4.14}$$

Las distancias se muestran

- $D_1 = \text{base} + \text{eslabón 1} = 0.95\text{m}$
- $L_2 = \text{eslabón 2} = 0.1\text{m}$
- $A_3 = \text{eslabón 3} = 0.5\text{m}$
- $L_5 = \text{eslabón 4} + \text{eslabón 5} = 0.4\text{m}$
- $L_7 = \text{eslabón 6} + \text{eslabón 7 (laparoscopio)} = 0.45\text{m}$

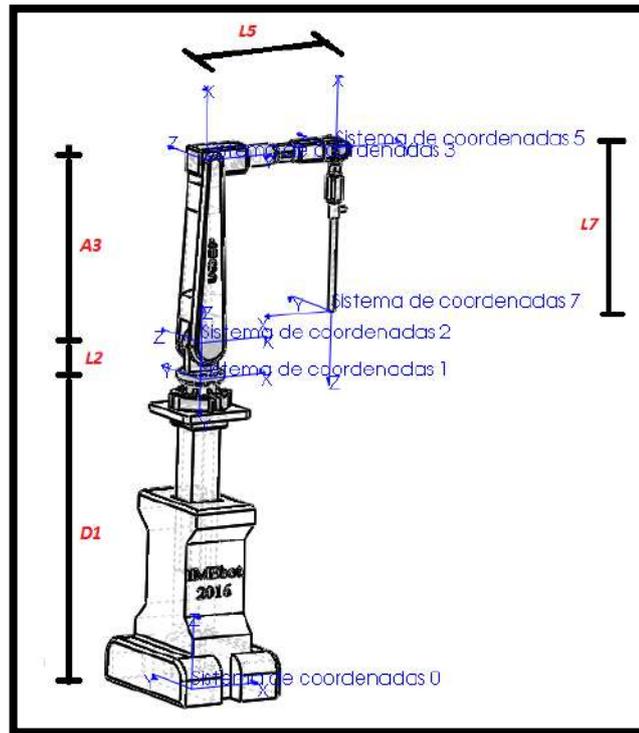


Figura 4.10 Valores numéricos con parámetros Denavit - Hartenberg (*elaboración propia*)

El modelo geométrico directo utilizando los parámetros Denavit – Hartenberg está dado por:

$$X = L5 * c2 * c34 - L7 * (c2 * c34 * s6 + c6 * s2 * s5 + c2 * c5 * c6 * s34) + A3 * c2 * s3 \quad (4.15)$$

$$Y = L5 * c34 * s2 - L7 * (c34 * s2 * s6 - c2 * c6 * s5 + c5 * c6 * s2 * s34) + A3 * s2 * s3 \quad (4.16)$$

$$Z = D1 + L2 + A3 * c3 - L5 * s34 + L7 * (s6 * s34 - c5 * c6 * c34) \quad (4.17)$$

La matriz de orientación 0A_7 tiene la forma general: ${}^0A_7 = \begin{bmatrix} n_x & o_x & a_x \\ n_y & o_y & a_y \\ n_z & o_z & a_z \end{bmatrix}$ cuando usamos

los parámetros Denavit - Hartenberg. Las ecuaciones que representan la orientación son:

$$n_x = c2 * s34 * (s5 * s7 + c5 * c7 * s6) - s2 * (c5 * s7 - c7 * s5 * s6) - c2 * c6 * c7 * c34 \quad (4.18)$$

$$n_y = c2 * (c5 * s7 - c7 * s5 * s6) + s2 * s34 * (s5 * s7 + c5 * c7 * s6) - c6 * c7 * c34 * s2 \quad (4.19)$$

$$n_z = c34 * (s5 * s7 + c5 * c7 * s6) + c6 * c7 * s34 \quad (4.20)$$

$$o_x = c2 * s34 * (c7 * s5 - c5 * s6 * s7) - s2 * (c5 * c7 + s5 * s6 * s7) + c2 * c6 * c34 * s7 \quad (4.21)$$

$$o_y = c2 * (c5 * c7 + s5 * s6 * s7) + s2 * s34 * (c7 * s5 - c5 * s6 * s7) + c6 * c34 * s2 * s7 \quad (4.22)$$

$$o_z = c34 * (c7 * s5 - c5 * s6 * s7) - c6 * s7 * s34 \quad (4.23)$$

$$a_x = -c2 * c34 * s6 - c6 * s2 * s5 - c2 * c5 * c6 * s34 \quad (4.24)$$

$$a_y = c2 * c6 * s5 - c34 * s2 * s6 - c5 * c6 * s2 * s34 \quad (4.25)$$

$$a_z = s6 * s34 - c5 * c6 * c34 \quad (4.26)$$

4.3.2 Modelamiento Geométrico Inverso (MGI)

El modelamiento geométrico inverso (MGI) calcula los valores de las variables articulares ($q_1, q_2, q_3, q_4, q_5, q_6, q_7$) de acuerdo a la orientación y posición actual o deseada del extremo distal del laparoscopio, expresado en el sistema coordenado (X, Y, Z).

El MGI tiene, en general, múltiples soluciones y su complejidad depende de la geometría del robot. En nuestro asistente IMEBot, se tiene 7 grados de libertad, incluyendo 1 prismática y 6 revolutas. Asumiremos el primer grado de libertad (q_1) constante, ya que se regulará manualmente de acuerdo a la altura de la camilla y de la postura del paciente.

Es necesario también, tomar en cuenta la restricción del paso por el trócar (que es el punto en el que se hace la incisión en el abdomen del paciente), ya que ésta limita los movimientos del robot. Este punto deberá ser fijo y se tomará sus coordenadas una vez el abdomen haya sido insuflado con CO₂. Esta ubicación es muy importante ya que nos ayudará a hallar los valores de algunas articulaciones.

En la Figura 4.12 se muestra el laparoscopio montado sobre el paciente en una sala de operaciones. La ubicación del robot es sólo referencial, ya que podría ubicarse también a la izquierda del paciente o ingresar por los pies, dependiendo el tipo de intervención.

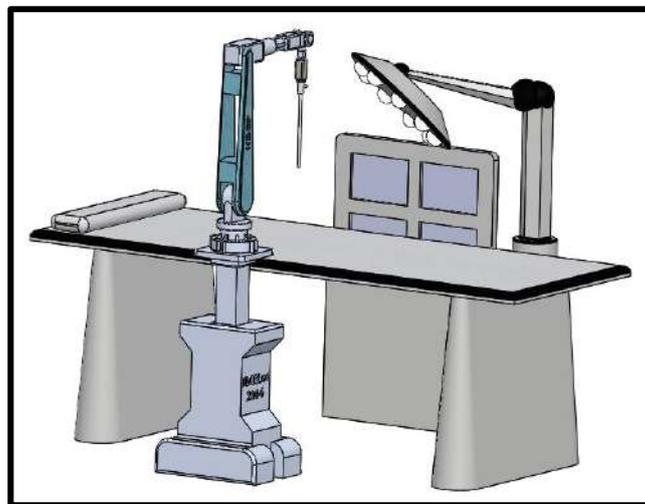


Figura 4.11 IMEBot montado en sala de operaciones 1 (*elaboración propia*)

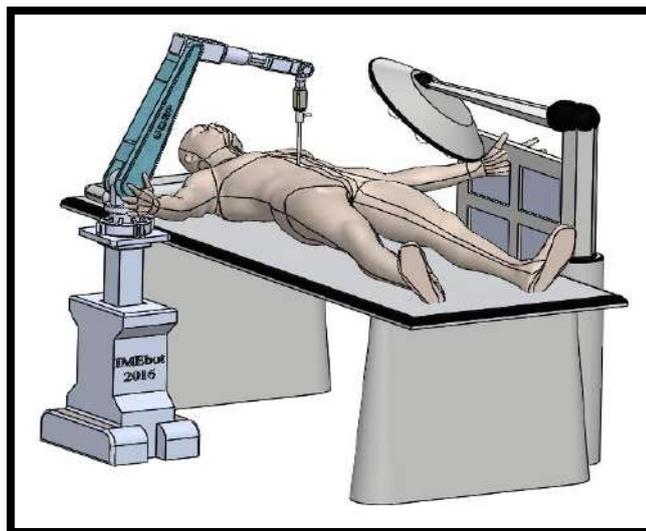


Figura 4.12 IMEbot montado en sala de operaciones 2 (*elaboración propia*)

Primero, se calcula el MGI utilizando las ecuaciones calculadas en el apartado 4.3.1.1 (parámetros Khalil - Kleinfinger) y se aplican el método de Paul y métodos numéricos. Posteriormente, se utiliza el apartado 3.5.2 (parámetros Denavit – Hartenberg) y se desacopla cinemáticamente el robot para el cálculo de los ángulos.

4.3.2.1 Método de Paul

Tomaremos como referencia los sistemas de coordenadas mostrados en la Figura 4.7. En nuestro análisis se verán afectados las posiciones de los sistemas 6 y 7, que son los que están ubicados antes y después del punto fijo de incisión. Entonces designaremos las posiciones cartesianas $P_6(x_6, y_6, z_6)$ y $P_7(x_7, y_7, z_7)$, mientras que el trócar estará ubicado en $P_t(x_t, y_t, z_t)$, tal como se muestra en la Figura 4.13.

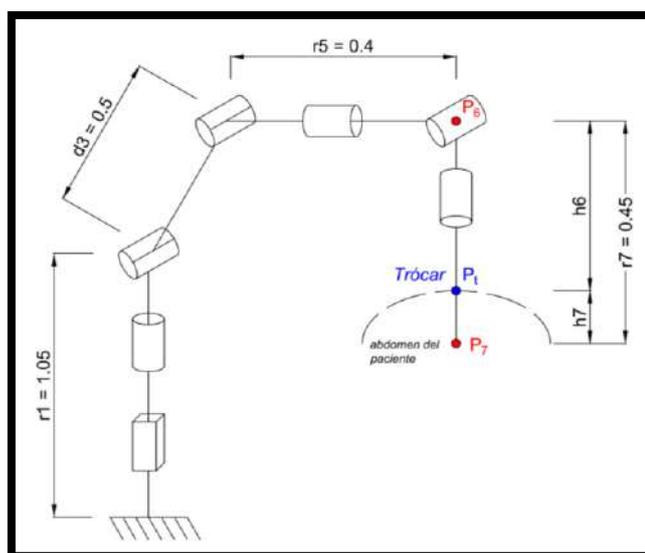


Figura 4.13 Movimiento alrededor del punto de incisión (*elaboración propia*)

Para incluir la restricción del movimiento consideraremos que el vector que va desde P_6 hasta P_7 , $\overline{P_6P_7}$, debe ser colineal con los vectores $\overline{P_6P_t}$ y $\overline{P_tP_7}$. Esto lo podemos representar como el producto vectorial de ambos vectores:

$$\overline{P_6P_t} \times \overline{P_tP_7} = 0 \quad (4.27)$$

$$(P_t - P_6) \times (P_7 - P_t) = 0 \quad (4.28)$$

También se debe tener en cuenta que la longitud de ambos vectores se debe mantener constante e igual a r_7 . Por lo tanto, la suma de los vectores $\overline{P_6P_t}$ y $\overline{P_tP_7}$ deberá ser igual a r_7 . Con esto aseguramos que el laparoscopio permanezca dentro del abdomen del paciente. Esto se expresa mediante la ecuación 4.30.

$$\overline{P_6P_t} + \overline{P_tP_7} = r_7 \quad (4.29)$$

$$|P_t - P_6| + |P_7 - P_t| = r_7 \quad (4.30)$$

$$h_6 + h_7 = r_7 \quad (4.31)$$

Con las ecuaciones necesarias para resolver el modelamiento geométrico inverso ya definidas, se procede a calcular la variable más lejana a la base del robot, es decir θ_7 .

Para esto se utilizará el método de Paul, que es un método iterativo y consiste en pre-multiplicar las matrices de transformación homogénea antes calculadas. En él definiremos una matriz U_0 con la localización y orientación deseada que no es, sino el modelamiento geométrico directo indicado anteriormente.

$$U_0 = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.32)$$

El método de Paul pre-multiplicamos sucesivamente las matrices ${}^{j-1}T_j$ a ambos lados de la ecuación 4.30.

$$U_0 = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 = {}^0T_7 = MGD \quad (4.33)$$

Entonces tenemos que:

$$U_1 = {}^1T_0 * U_0 = {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 \quad (4.34)$$

$$U_2 = {}^2T_1 {}^1T_0 * U_0 = {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 \quad (4.35)$$

$$U_3 = {}^3T_2 {}^2T_1 {}^1T_0 * U_0 = {}^3T_4 {}^4T_5 {}^5T_6 {}^6T_7 \quad (4.36)$$

$$U_4 = {}^4T_3 {}^3T_2 {}^2T_1 {}^1T_0 * U_0 = {}^4T_5 {}^5T_6 {}^6T_7 \quad (4.37)$$

$$U_5 = {}^5T_4 {}^4T_3 {}^3T_2 {}^2T_1 {}^1T_0 * U_0 = {}^5T_6 {}^6T_7 \quad (4.38)$$

$$U_6 = {}^6T_5 {}^5T_4 {}^4T_3 {}^3T_2 {}^2T_1 {}^1T_0 * U_0 = {}^6T_7 \quad (4.39)$$

Como para el MGI se conoce la posición y orientación final, se conocen los valores de U_0 , entonces, de la ecuación (4.39) los términos a la derecha e izquierda que representan la primera iteración U_6 del método de Paul son:

$$\begin{bmatrix} s_{x6} & n_{x6} & a_{x6} & P_{x6} \\ s_{y6} & n_{y6} & a_{y6} & P_{y6} \\ s_{z6} & n_{z6} & a_{z6} & P_{z6} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ 0 & 0 & -1 & -r_7 \\ s_7 & c_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.40)$$

Los valores calculados para s_{z6} y s_{x6} son:

$$\begin{aligned} s_{z6} &= s5 * (c4 * (s3 * sz + c3 * (c2 * sx + s2 * sy)) + s4 * (c3 * sz - s3 * (c2 * sx + s2 * sy))) - c5 * (c2 * sy - s2 * sx) \\ s_{x6} &= c6 * (c5 * (c4 * (s3 * sz + c3 * (c2 * sx + s2 * sy)) + s4 * (c3 * sz - s3 * (c2 * sx + s2 * sy))) + s5 * (c2 * sy - s2 * sx)) + s6 * (c4 * (c3 * sz - s3 * (c2 * sx + s2 * sy)) - s4 * (s3 * sz + c3 * (c2 * sx + s2 * sy))) \end{aligned}$$

En las ecuaciones anteriores se puede apreciar que los valores calculados están en función de las articulaciones $\theta_2, \theta_3, \theta_4, \theta_5, \theta_6$. Despejamos la variable θ_7 de (4.40):

$$\theta_7 = \arctan\left(\frac{s_{z6}}{s_{x6}}\right) \quad (4.41)$$

Ahora es necesario que la articulación θ_6 sea calculada incluyendo los límites de movimiento del trócar y la colinealidad con θ_7 . Para esto utilizaremos la ecuación (4.28), en la cual reemplazaremos los valores de P_6 y P_7 . La ubicación del trócar se considera conocida $P_t(t_x, t_y, t_z)$.

Los valores de P_7 son los calculados anteriormente en el MGD (ecuaciones 4.15, 4.16, 4.17), mientras que el punto P_6 se calculará a partir de la matriz 0T_6 :

$${}^0T_6 = \begin{bmatrix} {}^0A_6 & {}^0P_6 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.42)$$

El punto 0P_6 calculado es:

$${}^0P_6 = \begin{pmatrix} c2 * c3 * d3 - c2 * r5 * s34 \\ c3 * d3 * s2 - r5 * s2 * s34 \\ r1 + c34 * r5 + d3 * s3 \end{pmatrix} \quad (4.43)$$

Si efectuamos el producto cruz en (4.28):

$$(P_t - P_6) \times (P_7 - P_t) = 0$$

$$\begin{aligned}
& \begin{bmatrix} tx \\ ty \\ tz \end{bmatrix} - \begin{bmatrix} {}^0P_{6x} \\ {}^0P_{6y} \\ {}^0P_{6z} \end{bmatrix} * \begin{bmatrix} {}^0P_{7x} \\ {}^0P_{7y} \\ {}^0P_{7z} \end{bmatrix} - \begin{bmatrix} tx \\ ty \\ tz \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} i & j & k \\ (tx - {}^0P_{6x}) & (ty - {}^0P_{6y}) & (tz - {}^0P_{6z}) \\ ({}^0P_{7x} - tx) & ({}^0P_{7y} - ty) & ({}^0P_{7z} - tz) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\
& \begin{vmatrix} (ty - {}^0P_{6y}) & (tz - {}^0P_{6z}) \\ ({}^0P_{7y} - ty) & ({}^0P_{7z} - tz) \end{vmatrix} \vec{i} - \begin{vmatrix} (tx - {}^0P_{6x}) & (tz - {}^0P_{6z}) \\ ({}^0P_{7x} - tx) & ({}^0P_{7z} - tz) \end{vmatrix} \vec{j} + \begin{vmatrix} (tx - {}^0P_{6x}) & (ty - {}^0P_{6y}) \\ ({}^0P_{7x} - tx) & ({}^0P_{7y} - ty) \end{vmatrix} \vec{k} = 0 \\
& \begin{pmatrix} (ty - {}^0P_{6y})({}^0P_{7z} - tz) - (tz - {}^0P_{6z})({}^0P_{7y} - ty) \\ -((tx - {}^0P_{6x})({}^0P_{7z} - tz) - (tz - {}^0P_{6z})({}^0P_{7x} - tx)) \\ (tx - {}^0P_{6x})({}^0P_{7y} - ty) - (ty - {}^0P_{6y})({}^0P_{7x} - tx) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (4.44)
\end{aligned}$$

Los términos calculados para la ecuación anterior son:

$$\begin{aligned}
(1,1) = & ((c2 * r7 * s5 + c5 * c34 * r7 * s2) * (r1 - tz + c34 * r5 + d3 * s3) + c5 \\
& * r7 * s34 * (ty - c3 * d3 * s2 + r5 * s2 * s34)) * s6 \\
& + (r7 * s2 * s34 * (r1 - tz + c34 * r5 + d3 * s3) - c34 * r7 \\
& * (ty - c3 * d3 * s2 + r5 * s2 * s34)) * c6 = 0
\end{aligned}$$

$$\begin{aligned}
(2,1) = & ((r7 * s2 * s5 - c2 * c5 * c34 * r7) * (r1 - tz + c34 * r5 + d3 * s3) - c5 \\
& * r7 * s34 * (tx - c2 * c3 * d3 + c2 * r5 * s34)) * s6 \\
& + (c34 * r7 * (tx - c2 * c3 * d3 + c2 * r5 * s34) - c2 * r7 * s34 \\
& * (r1 - tz + c34 * r5 + d3 * s3)) * c6 = 0
\end{aligned}$$

$$\begin{aligned}
(3,1) = & ((c2 * r7 * s5 + c5 * c34 * r7 * s2) * (tx - c2 * c3 * d3 + c2 * r5 * s34) \\
& + (r7 * s2 * s5 - c2 * c5 * c34 * r7) * (ty - c3 * d3 * s2 + r5 * s2 \\
& * s34)) * s6 + (r7 * s2 * s34 * (tx - c2 * c3 * d3 + c2 * r5 * s34) \\
& - c2 * r7 * s34 * (ty - c3 * d3 * s2 + r5 * s2 * s34)) * c6 = 0
\end{aligned}$$

Utilizaremos las dos primeras filas (1,1) y (2,1) para calcular θ_6 . Construiremos un sistema de dos ecuaciones con dos incógnitas de la siguiente forma:

$$M_1 * s6 + N_1 * c6 = 0$$

$$M_2 * s6 + N_2 * c6 = 0$$

Resolviendo este sistema podremos calcular el ángulo de la articulación 6. Tenemos que:

$$\frac{s6}{c6} = \frac{N_2 - N_1}{M_1 - M_2}$$

$$\theta_6 = \arctan\left(\frac{s_6}{c_6}\right) \quad (4.45)$$

No utilizaremos la tercera fila de 4.42 ya que es linealmente dependiente de las dos primeras filas.

En la segunda iteración U_5 del método de Paul ecuación (4.38):

$$\begin{bmatrix} s_{x5} & n_{x5} & a_{x5} & P_{x5} \\ s_{y5} & n_{y5} & a_{y5} & P_{y5} \\ s_{z5} & n_{z5} & a_{z5} & P_{z5} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c_6 * c_7 & -c_6 * s_7 & s_6 & r_7 * s_6 \\ -s_7 & -c_7 & 0 & 0 \\ s_6 * c_7 & -s_6 * s_7 & -c_6 & -c_6 * r_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.46)$$

Nos permite obtener nuevamente los valores de los ángulos para las articulaciones 6 y 7, pudiendo verificar en este caso los resultados obtenidos previamente.

$$\theta_7 = \arctan\left(\frac{-s_{y5}}{-n_{y5}}\right) \quad (4.47)$$

$$\theta_6 = \arctan\left(\frac{a_{x5}}{-a_{z5}}\right) \quad (4.48)$$

En la tercera iteración U_4 , ecuación (4.37), tenemos:

$$\begin{bmatrix} s_{x4} & n_{x4} & a_{x4} & P_{x4} \\ s_{y4} & n_{y4} & a_{y4} & P_{y4} \\ s_{z4} & n_{z4} & a_{z4} & P_{z4} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_5 s_7 + c_5 c_6 c_7 & s_5 c_7 - c_5 c_6 s_7 & c_5 s_6 & c_5 s_6 r_7 \\ s_6 c_7 & -s_6 s_7 & -c_6 & r_5 - c_6 r_7 \\ c_5 s_7 - s_5 c_6 c_7 & c_5 c_7 + s_5 c_6 s_7 & -s_5 s_6 & -s_5 s_6 r_7 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.49)$$

$$\theta_5 = \arctan\left(\frac{-a_{z4}}{a_{x4}}\right) \quad (4.50)$$

Donde los términos de la izquierda están en función de los ángulos $\theta_2, \theta_3, \theta_4$, siendo:

$$a_{z4} = ax * s2 - ay * c2$$

$$a_{x4} = c4 * (az * s3 + c3 * (ax * c2 + ay * s2)) + s4 * (az * c3 - s3 * (ax * c2 + ay * s2))$$

Para la cuarta iteración U_3 , ecuación (4.36) y que se muestra en su forma expandida en el anexo 9.1, observamos que en los componentes de las matrices encontradas para el cálculo de θ_4 , no es posible despejar la variable, ya que estarán en función de $\theta_5, \theta_6, \theta_7$. Por lo tanto utilizaremos la ecuación (4.30), que garantizará que la parte del laparoscopio dentro del abdomen sumado a la parte exterior sea igual a r_7 .

Si tenemos el punto P_6 que está expresado en (4.43), P_7 en la ecuación (4.32) y P_t que es conocido, entonces

$$|P_t - P_6| + |P_7 - P_t| = r_7$$

$$\left(\begin{array}{c} tx - P_{6x} \\ ty - P_{6y} \\ tz - P_{6z} \end{array} \right) + \left(\begin{array}{c} P_x - tx \\ P_y - ty \\ P_z - tz \end{array} \right) = r_7$$

$$(tx - P_{6x})^2 + (ty - P_{6y})^2 + (tz - P_{6z})^2 = (r_7 - |P_7 - P_t|)^2 \quad (4.51)$$

Teniendo en cuenta la ecuación trigonométrica: $a * \text{sen}\theta + b * \text{cos}\theta = c$

Desarrollamos, factorizamos la ecuación (4.51) y le damos la forma anterior, para poder expresarla como:

$$a * \sin \theta_4 + b * \cos \theta_4 = c \quad (4.52)$$

Donde:

- $\mathbf{a} = r_5 * (c_3 * (2 * c_2 * (tx - c_2 * c_3 * d_3) + 2 * s_2 * (ty - c_3 * d_3 * s_2)) - 2 * s_3 * (r_1 - tz + d_3 * s_3))$
- $\mathbf{b} = r_5 * (s_3 * (2 * c_2 * (tx - c_2 * c_3 * d_3) + 2 * s_2 * (ty - c_3 * d_3 * s_2)) + 2 * c_3 * (r_1 - tz + d_3 * s_3))$
- $\mathbf{c} = (r_7 - ((P_x - tx)^2 + (P_y - ty)^2 + (P_z - tz)^2)^{(1/2)})^2 - (tx - c_2 * c_3 * d_3)^2 - (ty - c_3 * d_3 * s_2)^2 - r_5^2 - (r_1 - tz + d_3 * s_3)^2$

Cuya solución se muestra en la tabla 3.1:

$$\theta_4 = \text{atan2}(a, b) \mp \text{atan2}(\sqrt{a^2 + b^2 - c^2}, c) \quad (4.53)$$

Para la quinta iteración U_2 , ecuación (4.35), podemos despejar la variable θ_2 tomando los términos $U_2(2,3)$, $U_2(2,4)$ y ${}^2T_7(2,3)$, ${}^2T_7(2,4)$. Igualamos términos y obtenemos:

$$a_1 * \sin \theta_2 + b_1 * \cos \theta_2 = 0$$

Donde:

- $\mathbf{a}_1 = r_7 * ax - Px$
- $\mathbf{b}_1 = Py - r_7 * ay$

$$\theta_2 = \text{atan2}(-b_1, a_1) \text{ ó } \theta_2 = \text{atan2}(b_1, -a_1) \quad (4.54)$$

En la sexta iteración U_1 , ecuación (4.34), podemos notar que de igual manera a la cuarta iteración U_3 , no podemos despejar r_1 ó θ_3 . Por lo tanto, de estas 2 iteraciones extraeremos algunas relaciones que nos permitan obtener θ_3 y consideraremos $r_1 = 1.05$ fijo y conocido. Entonces:

$$U_1(3,3) = c_5 * s_6 * s_34 - 1.0 * c_6 * c_34 = {}^1T_7(3,3) = az$$

$$U_1(3,4) = d_3 * s_3 + c_34 * (r_5 - c_6 * r_7) + c_5 * r_7 * s_6 * s_34 = {}^1T_7(3,4) \\ = Pz - r_1$$

Reemplazamos $U_1(3,3)$ en $U_1(3,4)$, tenemos:

$$(Pz - r1 - r7 * az) = d3 * s3 + c34 * r5 \quad (4.55)$$

Utilizaremos también de la matriz U_3 :

$$U_3(2,3) = az * c3 - s3 * (ax * c2 + ay * s2) = {}^3T_7(2,3) = c5 * s4 * s6 - c4 * c6$$

$$U_3(2,4) = c3 * (Pz - r1) - s3 * (Px * c2 + Py * s2) = {}^3T_7(2,4) \\ = c4 * r5 - r7 * (c4 * c6 - c5 * s4 * s6)$$

Reemplazamos $U_3(2,3)$ en $U_3(2,4)$, tenemos:

$$c3(Pz - r1 - r7 * az) + s3(-Px * c2 - Py * s2 + r7 * ax * c2 \\ + r7 * ay * s2) = c4 * r5 \quad (4.56)$$

Ahora (4.55) en (4.56), obtenemos:

$$a_2 * \sin \theta_4 + b_2 * \cos \theta_4 = c_2 \quad (4.57)$$

Donde:

- $a_2 = -c3 * s3 * r5$
- $b_2 = r5 * s3^2$
- $c_2 = s3 * (Px * c2 + Py * s2 - r7 * ax * c2 - r7 * ay * s2 - c3 * d3)$

$$\theta_4 = \text{atan2}(a_2, b_2) \mp \text{atan2}(\sqrt{a_2^2 + b_2^2 - c_2^2}, c_2) \quad (4.58)$$

Si comparamos la ecuación (4.58) con la hallada anteriormente en (4.53), podemos notar que tenemos dos ecuaciones distintas que nos permiten calcular θ_4 . Ambas ecuaciones están en función de componentes conocidos y de la articulación θ_3 , por lo que se utilizarán para su cálculo.

Aplicaremos entonces un *método numérico* para calcular θ_3 . Se empleará el método de la bisección. El ángulo θ_3 tendrá que estar comprendido en el intervalo $[10^\circ, 115^\circ]$ debido al rango de trabajo que se propondrá. Éstos serán los valores $[xa, xb]$ del intervalo a ingresar en el procedimiento iterativo, tal y como se implementó en la función en MATLAB que se muestra en el anexo 9.2.2.1.2.

La resolución y programación de las ecuaciones hechas en Matlab para el modelamiento inverso se muestran en el archivo 'MGI.m' en el anexo 9.2.2.1.1. Cabe mencionar que los ángulos θ_2 y θ_4 daban dos posibles soluciones cada uno, sin embargo, se omitieron los resultados que daban una configuración inadecuada del robot (es decir, la configuración codo abajo). Los valores encontrados permiten posicionar a IMEbot de tal forma que mantiene un punto de pivote fijo en la posición del trócar.

Resumiendo, se muestra a continuación el diagrama esquemático del cálculo de las variables utilizando en este apartado.

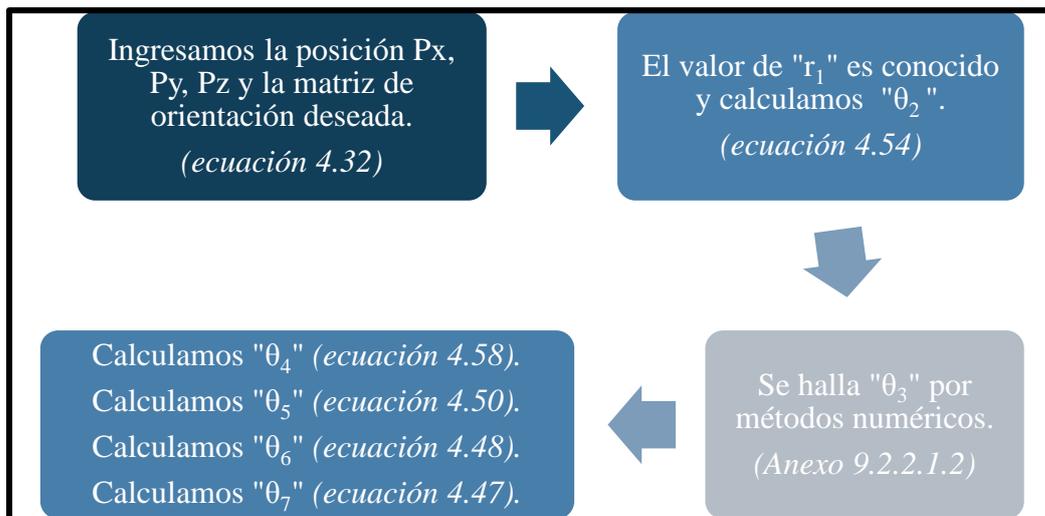


Figura 4.14 Cálculo del MGI mediante Método de Paul

Es importante mencionar que el uso de este método numérico para el cálculo es relativamente lento, ya que al tratar de encontrar el ángulo de una forma iterativa, la ejecución del algoritmo en Matlab toma un aproximado de 2.5 segundos. Es por esta razón que en el siguiente apartado se trata de encontrar todos los valores angulares a partir de ecuaciones algebraicas directas.

4.3.2.2 Desacoplo Cinemático

En este epígrafe utilizaremos las ecuaciones y transformaciones calculadas utilizando los parámetros Denavit – Hartenberg. Dado que las últimas tres articulaciones de IMEbot son las encargadas de la orientación y sus ejes se cortan en un punto es posible utilizar el método de desacoplo cinemático.

En este método dado una posición y orientación final deseada, se calcula las coordenadas del punto de corte, también llamado *muñeca del robot*, y se calculan las primeras tres variables encargadas de posicionar este punto \vec{p}_m (que en nuestro caso son q_2, q_3, q_4). Una vez calculados éstos, se obtiene el resto de las variables articulares.

El punto central \vec{p}_m corresponde a la intersección de los ejes $\mathbf{z}_5, \mathbf{z}_6, \mathbf{z}_7$, siendo el origen de los sistemas $\{S_5\}, \{S_6\}$ (Figura 4.8 y Figura 4.9). Consideraremos también la posición del extremo del robot como \vec{P}_7 . Es decir:

$$\vec{p}_m = \vec{O_0O_6} \quad (4.59)$$

$$\vec{P}_7 = \vec{O_0O_7} \quad (4.60)$$

Si de acuerdo al algoritmo de Denavit – Hartenberg (DH9) la dirección de los ejes \mathbf{z}_6 y \mathbf{z}_7 debe ser la misma, entonces la distancia de los orígenes O_6 y O_7 medida a lo largo de \mathbf{z}_7 es justamente L_7 (ver figura 4.10). Entonces:

$$\vec{p}_m = \vec{P}_7 - L_7 * \vec{z}_7 \quad (4.61)$$

Donde \mathbf{z}_7 es un vector unitario y puede ser calculado a partir de las posiciones conocidas del extremo final del laparoscopio $\vec{P}_7(P_x, P_y, P_z)$ y de la ubicación del trócar en la pared abdominal $\vec{P}_t(t_x, t_y, t_z)$:

$$\vec{z}_7 = \frac{\vec{P}_7 - \vec{P}_t}{\|\vec{P}_7 - \vec{P}_t\|} \quad (4.62)$$

La base y los eslabones 1, 2, 3, 4 y 5 finalizan en p_m . Si consideramos en las variables articulares que $q_1 = D1$ es fijo (se regula acorde a la altura de la camilla), y tanto q_4 como q_5 no modifican la posición de $\vec{p}_m(x_m, y_m, z_m)$, entonces podemos calcular q_2, q_3, q_4 por métodos geométricos. Para esto consideraremos la configuración codo arriba y las variables involucradas en el cálculo que se muestran en la Figura 4.15.

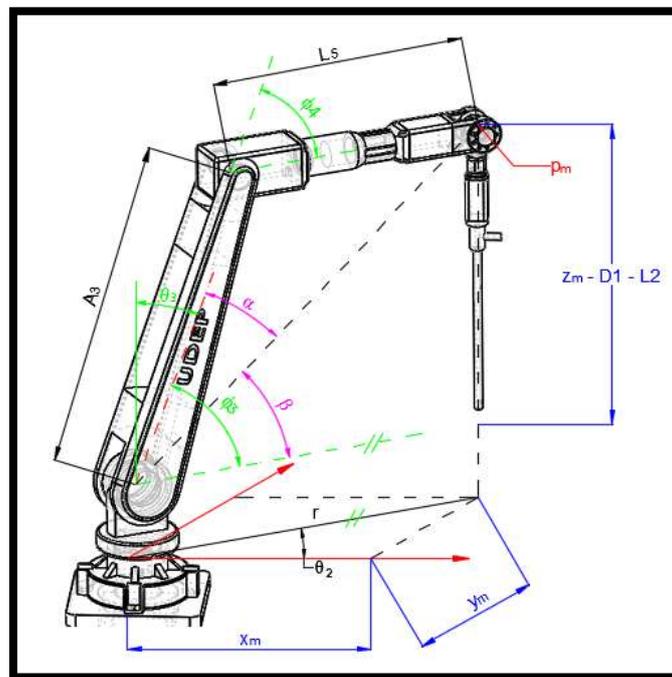


Figura 4.15 Desacoplo cinemático IMEbot (elaboración propia)

Como se aprecia, esta sub-robot tiene una estructura planar, quedando definido el ángulo θ_2 .

$$\theta_2 = \arctan\left(\frac{y_m}{x_m}\right) \quad (4.63)$$

Considerando ahora solamente las longitudes A_3 (eslabón 3) y L_5 (eslabón 4 + eslabón 5) que están situados en un mismo plano, se tendrá:

$$r^2 = x_m^2 + y_m^2$$

$$r^2 + (z_m - D1 - L2)^2 = A3^2 + L5^2 + 2 * A3 * L5 * \cos(\phi_4)$$

$$\cos(\phi_4) = \frac{r^2 + (z_m - D1 - L2)^2 - A3^2 - L5^2}{2 * A3 * L5} \quad (4.64)$$

Puesto que

$$\sin(\phi_4) = \pm \sqrt{1 - \cos^2(\phi_4)}$$

Tendremos:

$$\phi_4 = \arctan\left(\frac{\pm \sqrt{1 - \cos^2(\phi_4)}}{\cos(\phi_4)}\right) \quad (4.65)$$

Debido a la asignación de sistemas de coordenadas Denavit – Hartenberg y a la configuración inicial deseada que posee el robot, ϕ_4 está relacionado con el ángulo θ_4 por:

$$\theta_4 = \phi_4 - 90^\circ \quad (4.66)$$

Como la configuración es codo arriba, para el cálculo de ϕ_3 y posteriormente θ_3 consideraremos los ángulos α y β mostrados en la figura 4.14, donde:

$$\alpha = \arctan\left(\frac{L5 * \sin(\phi_4)}{A3 + L5 * \cos(\phi_4)}\right) \quad (4.67)$$

$$\beta = \arctan\left(\frac{z_m - D1 - L2}{r = \pm \sqrt{x_m^2 + y_m^2}}\right) \quad (4.68)$$

Si $\phi_3 = \alpha + \beta$, entonces de la misma forma que en (4.66), tendremos:

$$\theta_3 = 90^\circ - \phi_3 \quad (4.69)$$

Una vez obtenidos los valores de $\theta_2, \theta_3, \theta_4$ podemos relacionarlos con las variables articulares globales q_2, q_3, q_4 (ver tabla 4.2). Ahora, debemos calcular θ_5, θ_6 y θ_7 que consiguen la orientación deseada. Para ello obtenemos la submatriz de rotación 0R_7 de 0T_7 :

$${}^0R_7 = [\mathbf{n} \ \mathbf{o} \ \mathbf{a}] = {}^0R_4 * {}^4R_7 \quad (4.70)$$

$${}^4R_7 = ({}^0R_4)^{-1} * {}^0R_7 \quad (4.71)$$

Donde: ${}^0R_4 = {}^0R_1 {}^1R_2 {}^2R_3 {}^3R_4$ y $({}^0R_4)^{-1} = ({}^0R_4)^T$ por ser matrices ortonormales. Entonces se obtiene:

$${}^0R_4 = \begin{bmatrix} -c2 * s34, & -s2 & c2 * c34 \\ -s2 * s34, & c2 & c34 * s2 \\ -c34, & 0 & -s34 \end{bmatrix}$$

Calculando ambos lados de la ecuación (4.73):

$${}^4R_7 = \begin{bmatrix} -s5 * s7 - c5 * c7 * c6, & c5 * s6 * s7 - c7 * s5 & c5 * c6 \\ c5 * s7 - c7 * s5 * s6, & c5 * c7 + s5 * s6 * s7 & c6 * s5 \\ -c6 * c7, & c6 * s7 & -s6 \end{bmatrix} \quad (4.72)$$

$$({}^0R_4)^{-1} * {}^0R_7 = Aux \quad (4.73)$$

$$= \begin{bmatrix} -c2 * s34 * nx - s2 * s34 * ny - c34 * nz & -c2 * s34 * ox - s2 * s34 * oy - c34 * oz & Aux(1,3) \\ -s2 * nx + c2 * ny & -s2 * ox + c2 * oy & Aux(2,3) \\ c2 * c34 * nx + c34 * s2 * ny - s34 * nz & c2 * c34 * ox + c34 * s2 * oy - s34 * oz & Aux(3,3) \end{bmatrix}$$

Donde:

$$Aux(1,3) = -c2 * s34 * ax - s2 * s34 * ay - c34 * az$$

$$Aux(2,3) = -s2 * ax + c2 * ay$$

$$Aux(3,3) = c2 * c34 * ax + c34 * s2 * ay - s34 * az$$

Debemos notar que el vector \vec{a} de la ecuación (4.70) es igual al vector unitario \vec{z}_7 calculado en (4.62) por lo que se conoce a priori. Entonces, utilizaremos sólo la tercera columna de las ecuaciones (4.72) y (4.73) para el cálculo de los ángulos. Éstos quedarán expresados como:

$$\theta_5 = \arctan\left(\frac{-s2 * ax + c2 * ay}{A3 + L5 * \cos(\phi_4)}\right) \quad (4.74)$$

$$\theta_6 = \arcsin(-c2 * c34 * ax - c34 * s2 * ay + s34 * az) \quad (4.75)$$

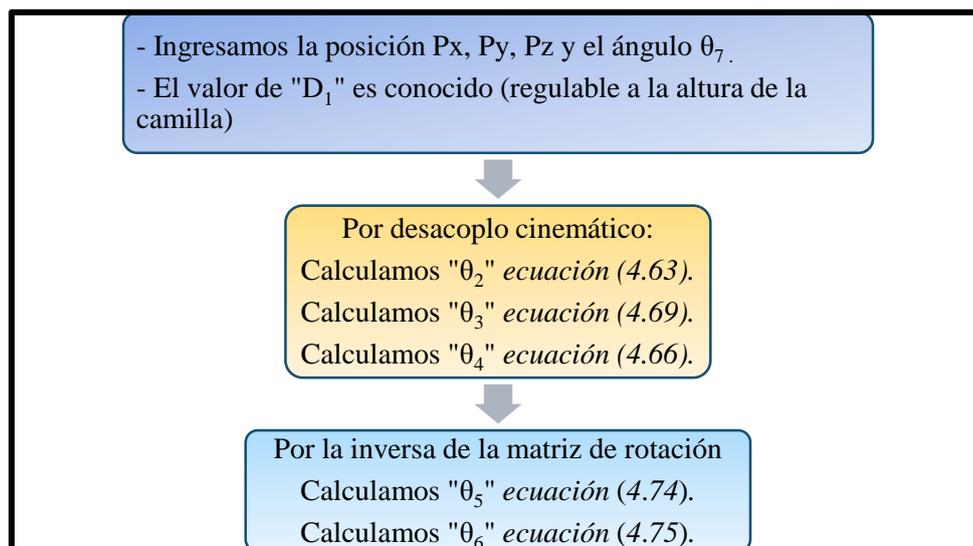


Figura 4.16 Cálculo del MGI mediante desacoplo cinemático

Podemos llegar a la conclusión de que para el cálculo de los seis primeros grados de libertad no es necesario especificar la matriz de orientación final deseada, sino basta con conocer la

posición final deseada y la ubicación del trócar. El último ángulo, es decir θ_7 , se debería ingresar directamente ya que es el encargado del giro del laparoscopio sobre su mismo eje. La variación de éste ángulo sólo es necesario en laparoscopios distintos al de 0° .

4.3.1 Prueba del Modelo Geométrico

El primer paso para comprobar nuestros modelos geométricos es la creación de una trayectoria arbitraria. Además se deberá corroborar que el paso por el trócar es correcto.

Dado que nuestro espacio de trabajo es el interior del abdomen, se dibuja una esfera, de tal manera que pueda existir movimiento en todas las articulaciones. Ésta tiene un radio de 5 cm y su centro tiene como posición (0.5, 0, 0.9) [metros] respecto a la base del robot. Se ha tomado en cuenta las dimensiones de la camilla y altura promedio del abdomen de una persona

Se asumirá que:

- Se ha ajustado previamente la altura del robot mediante la articulación 1.
- La posición inicial del robot es en la parte superior de la esfera.
- La velocidad en el instante inicial es 0.
- se usa un laparoscopio de 0° y por tanto no es necesario el movimiento en la articulación 7. Sin embargo, los posteriores análisis dejan libre la posibilidad de cambiar este valor.

La trayectoria construida contiene 340 puntos, un recorrido lineal de 3.791m y empieza su transición desde la parte superior [0.5m; 0; 0.95m].

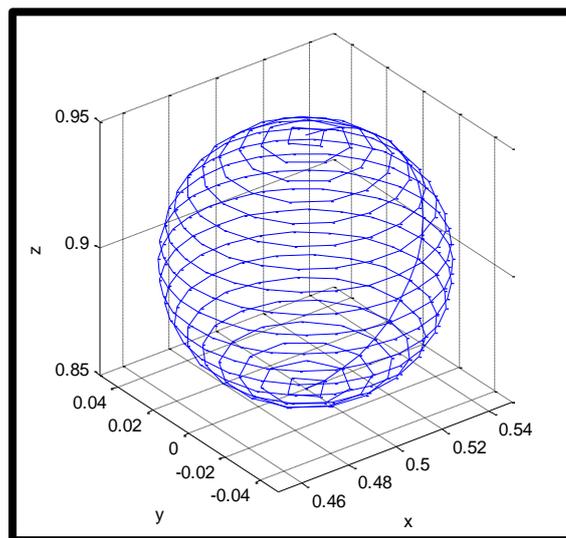


Figura 4.17 Trayectoria esférica

El diagrama de bloques implementado en Simulink, conteniendo las funciones con los modelos geométricos directo e inverso se ve a continuación:

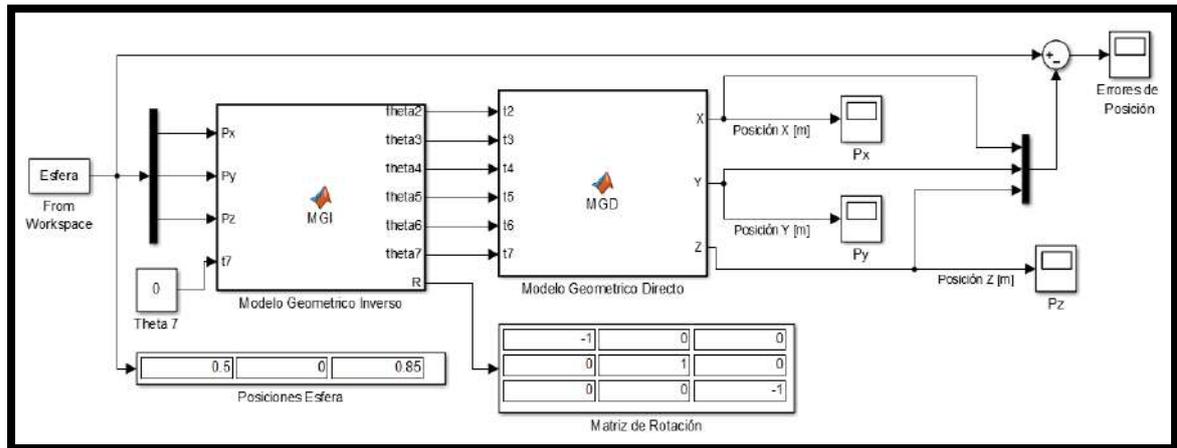


Figura 4.18 Prueba Modelo Geométrico

Se puede apreciar en la Figura 4.18 que se ingresa la trayectoria 'Esfera'. La función 'Modelo Geométrico Inverso' permite obtener los ángulos articulares, mientras que 'Modelo Geométrico Directo' vuelve a convertir estos ángulos en posiciones cartesianas. La posición final de la esfera es [0.5m; 0m; 0.85m] y la matriz de orientación final queda correctamente descrita, es decir, el laparoscopio queda fijado verticalmente hacia abajo.

Las trayectorias en el espacio cartesiano para cada eje se muestran en la figura Figura 4.19. Las posiciones en el eje 'x' varían entre [-0.45, 0.55], en el eje 'y' entre [-0.05, 0.05], mientras que en el eje 'z' decrece entre [0.95, 0.85].

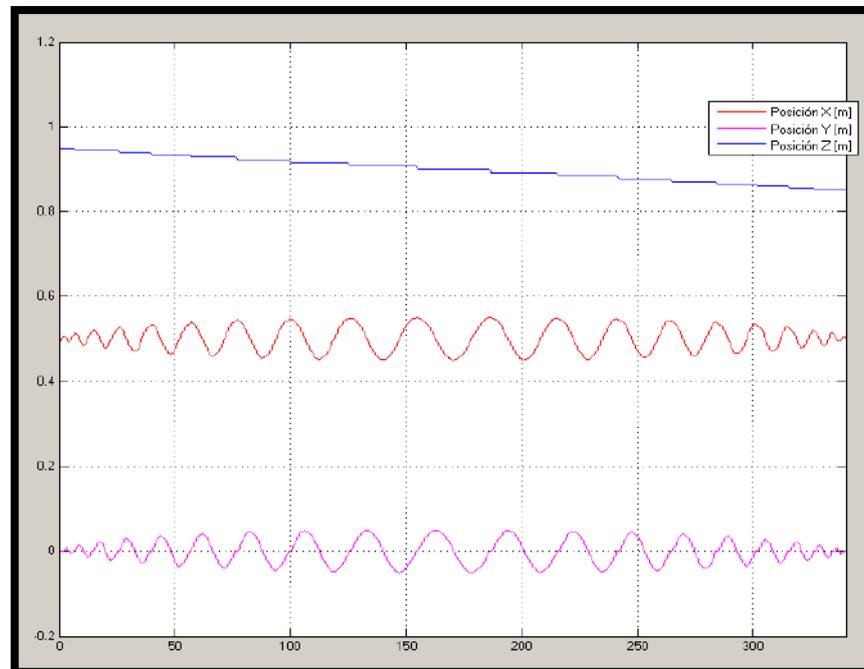


Figura 4.19 Posiciones cartesianas para describir una trayectoria esférica

Las posiciones articulares para lograr dichas posiciones cartesianas se muestran a continuación.

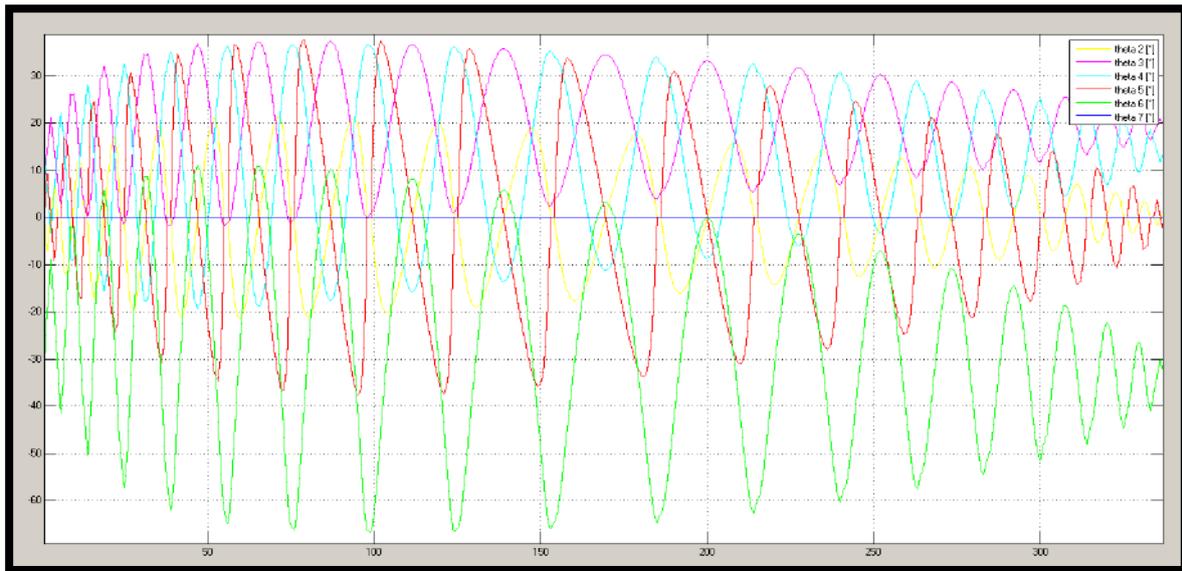


Figura 4.20 Posiciones articulares para describir una trayectoria esférica

Se puede notar que la trayectoria de las articulaciones 3,4 y 6 no son simétricas y van aumentando o disminuyendo conforme se requiera. Esto se debe a que mediante el MGI se calculan ángulos que respeten el paso por el trócar.

El error en posición producido por los modelos geométricos es bastante aceptable ya que son del orden de $5e-16$.

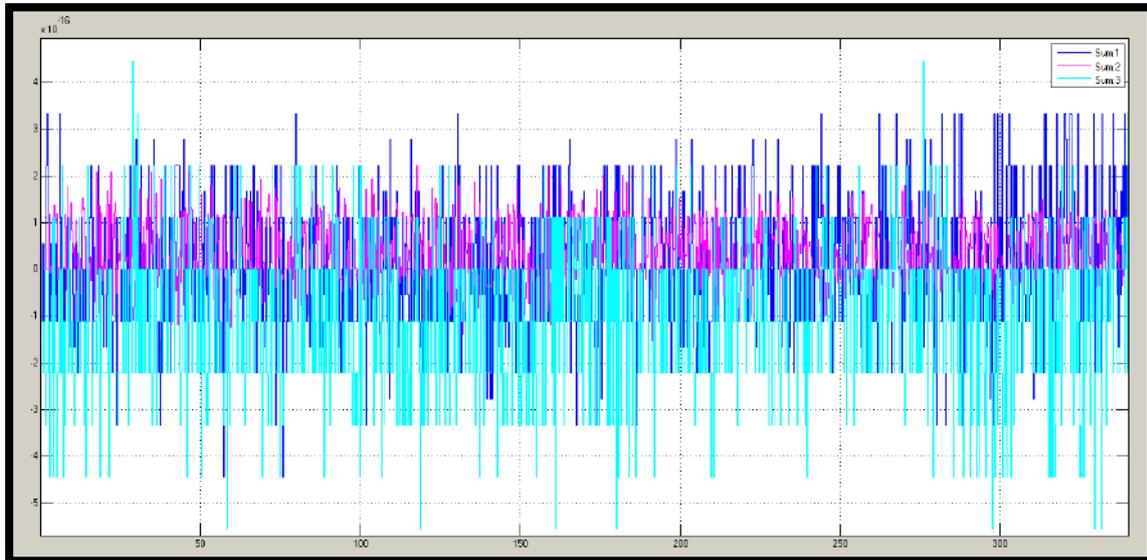


Figura 4.21 Error de posición por el uso de los modelos geométricos

4.4 GENERACIÓN DE TRAYECTORIA

En el apartado anterior se construyó una trayectoria esférica, sin embargo, es necesario tomar en cuenta ciertos criterios de calidad de trayectoria, como suavidad o precisión y además que cumpla con los requerimientos de velocidad en cada punto seguido.

Por lo tanto se implementó una función en Matlab ‘Interpolar.m’, que permite interpolar los puntos obtenidos anteriormente y generar para cada articulación una trayectoria realizable por sus actuadores. Para ello se utilizó una función polinómica ‘spline_cúbico.m’, el cual emplea cuatro condiciones de contorno: posición y velocidad en cada intervalo, es decir de punto a punto. La trayectoria que une dos puntos adyacentes (q^i, q^{i+1}) es ^[31]:

$$q(t) = a + b(t - t^i) + c(t - t^i)^2 + d(t - t^i)^3 \quad (4.76)$$

Donde

$$a = q^i$$

$$b = q^{i+1}$$

$$T = t^{i+1} - t^i$$

$$c = \frac{3}{T^2}(q^{i+1} - q^i) - \frac{1}{T}(\dot{q}^{i+1} - \dot{q}^i)$$

$$d = -\frac{2}{T^3}(q^{i+1} - q^i) + \frac{1}{T^2}(\dot{q}^{i+1} + \dot{q}^i)$$

4.5 MODELAMIENTO CINEMÁTICO IMEBOT

Para conocer la velocidad de la posición y orientación del extremo distal del laparoscopio a partir de las velocidades articulares $(\dot{q}_1, \dot{q}_2, \dot{q}_3, \dot{q}_4, \dot{q}_5, \dot{q}_6, \dot{q}_7)$, es necesario obtener el Jacobiano geométrico. Existen diferentes procedimientos para su obtención como se mencionó en el apartado 3.7.3, y su importancia implica la coordinación entre la velocidad lineal y angular del laparoscopio con la de las articulaciones.

Se ha realizado la obtención del modelo geométrico basado en la propagación de velocidades, el cual permite obtener las columnas de la matriz Jacobiana.

El primer paso es obtener los vectores directores, que están definidos como ${}^0\mathbf{z}_i = {}^0\mathbf{T}_i(1:3,3)$. Por lo tanto:

$$\begin{aligned} {}^0\mathbf{z}_0 = {}^0\mathbf{z}_1 &= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}; & {}^0\mathbf{z}_2 = {}^0\mathbf{z}_3 &= \begin{pmatrix} -s2 \\ c2 \\ 0 \end{pmatrix}; \\ {}^0\mathbf{z}_4 &= \begin{pmatrix} c2 * c34 \\ s2 * c34 \\ -s34 \end{pmatrix}; & {}^0\mathbf{z}_5 &= \begin{pmatrix} c2 * s34 * s5 - s2 * c5 \\ s2 * s34 * s5 + c2 * c5 \\ c34 * s5 \end{pmatrix}; \\ {}^0\mathbf{z}_6 = {}^0\mathbf{z}_7 &= \begin{pmatrix} -c2 * c34 * s6 - s2 * s5 * c6 - c2 * s34 * c5 * c6 \\ -s2 * c34 * s6 + c2 * s5 * c6 - s2 * s34 * c5 * c6 \\ s34 * s6 - c5 * c6 * c34 \end{pmatrix}; \end{aligned}$$

Posteriormente, se calcula los vectores que van desde el origen del sistema $\{S_i\}$ hasta el extremo del robot $\{S_n\}$, los cuales están determinados por ${}^i\mathbf{p}_n = {}^0\mathbf{T}_n(1:3,4) - {}^0\mathbf{T}_i(1:3,4)$.

$${}^1\mathbf{p}_7 = \begin{pmatrix} L5 * c2 * c34 - L7 * (c2 * c34 * s6 + c6 * s2 * s5 + c2 * c5 * c6 * s34) + A3 * c2 * s3 \\ L5 * c34 * s2 - L7 * (c34 * s2 * s6 - c2 * c6 * s5 + c5 * c6 * s2 * s34) + A3 * s2 * s3 \\ L2 + A3 * c3 - L5 * s34 + L7 * (s6 * s34 - c5 * c6 * c34) \end{pmatrix}$$

$${}^2\mathbf{p}_7 = \begin{pmatrix} L5 * c2 * c34 - L7 * (c2 * c34 * s6 + c6 * s2 * s5 + c2 * c5 * c6 * s34) + A3 * c2 * s3 \\ L5 * c34 * s2 - L7 * (c34 * s2 * s6 - c2 * c6 * s5 + c5 * c6 * s2 * s34) + A3 * s2 * s3 \\ A3 * c3 - L5 * s34 + L7 * (s6 * s34 - c5 * c6 * c34) \end{pmatrix}$$

$${}^3\mathbf{p}_7 = {}^4\mathbf{p}_7 = \begin{pmatrix} L5 * c2 * c34 - L7 * (c2 * c34 * s6 + c6 * s2 * s5 + c2 * c5 * c6 * s34) \\ L5 * c34 * s2 - L7 * (c34 * s2 * s6 - c2 * c6 * s5 + c5 * c6 * s2 * s34) \\ -L5 * s34 + L7 * (s6 * s34 - c5 * c6 * c34) \end{pmatrix}$$

$${}^5\mathbf{p}_7 = {}^6\mathbf{p}_7 = \begin{pmatrix} -L7 * (c2 * c34 * s6 + c6 * s2 * s5 + c2 * c5 * c6 * s34) \\ -L7 * (c34 * s2 * s6 - c2 * c6 * s5 + c5 * c6 * s2 * s34) \\ L7 * (s6 * s34 - c5 * c6 * c34) \end{pmatrix}$$

Finalmente, definidos los vectores ${}^0\mathbf{z}_i$ y ${}^i\mathbf{p}_n$, se puede obtener la matriz Jacobiana de acuerdo a la ecuación 3.15. Entonces:

$$J_1 = (0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0)^T$$

$$J_2 = \begin{pmatrix} L7 * (c34 * s2 * s6 - c2 * c6 * s5 + c5 * c6 * s2 * s34) - L5 * c34 * s2 - A3 * s2 * s3 \\ L5 * c2 * c34 - L7 * (c2 * c34 * s6 + c6 * s2 * s5 + c2 * c5 * c6 * s34) + A3 * c2 * s3 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$J_3 = \begin{pmatrix} L7 * (c2 * s6 * s34 - c2 * c5 * c6 * c34) + A3 * c2 * c3 - L5 * c2 * s34 \\ L7 * (s2 * s6 * s34 - c5 * c6 * c34 * s2) + A3 * c3 * s2 - L5 * s2 * s34 \\ L7 * (c34 * s6 + c5 * c6 * s34) - L5 * c34 - A3 * s3 \\ -s2 \\ c2 \\ 0 \end{pmatrix}$$

$$J_4 = \begin{pmatrix} L7 * (c2 * s6 * s34 - c2 * c5 * c6 * c34) - L5 * c2 * s34 \\ L7 * (s2 * s6 * s34 - c5 * c6 * c34 * s2) - L5 * s2 * s34 \\ L7 * (c34 * s6 + c5 * c6 * s34) - L5 * c34 \\ -s2 \\ c2 \\ 0 \end{pmatrix}$$

$$J_5 = \begin{pmatrix} L7 * (c2 * c6 * s5 * s34 - c5 * c6 * s2) \\ L7 * (c6 * s2 * s5 * s34 + c2 * c5 * c6) \\ L7 * c34 * c6 * s5 \\ c2 * s34 \\ s2 * c34 \\ -s34 \end{pmatrix}$$

$$J_6 = \begin{pmatrix} L7 * (s2 * s5 * s6 * c34^2 - c2 * c6 * c34 + s2 * s5 * s6 * s34^2 + c2 * c5 * s6 * s34) \\ -L7 * (c2 * c34^2 * s5 * s6 - c5 * s2 * s6 * s34 + c2 * s5 * s6 * s34^2 + c6 * c34 * s2) \\ L7 * (c6 * s34 + c5 * c34 * s6) \\ c2 * s5 * s34 - c5 * s2 \\ c2 * c5 + s2 * s5 * s34 \\ c34 * s5 \end{pmatrix}$$

$$J_7 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -c2 * c34 * s6 - c6 * s2 * s5 - c2 * c5 * c6 * s34 \\ c2 * c6 * s5 - c34 * s2 * s6 - c5 * c6 * s2 * s34 \\ s6 * s34 - c5 * c6 * c34 \end{pmatrix}$$

Sin embargo, debido a que la primera articulación no presentará movimiento, es decir, $q_1 = \dot{q}_1 = \ddot{q}_1 = 0$, podemos no incluirla en el Jacobiano y así tener una matriz cuadrada 6x6, que será invertible y permitirá el cálculo del modelo cinemático directo e inverso.

Entonces, finalmente tendremos:

$$J = [J_2 | J_3 | J_4 | J_5 | J_6 | J_7] \quad (4.77)$$

El modelo cinemático implementado en Simulink se muestra a continuación y la función implementada se puede ver en el anexo 9.2.5.

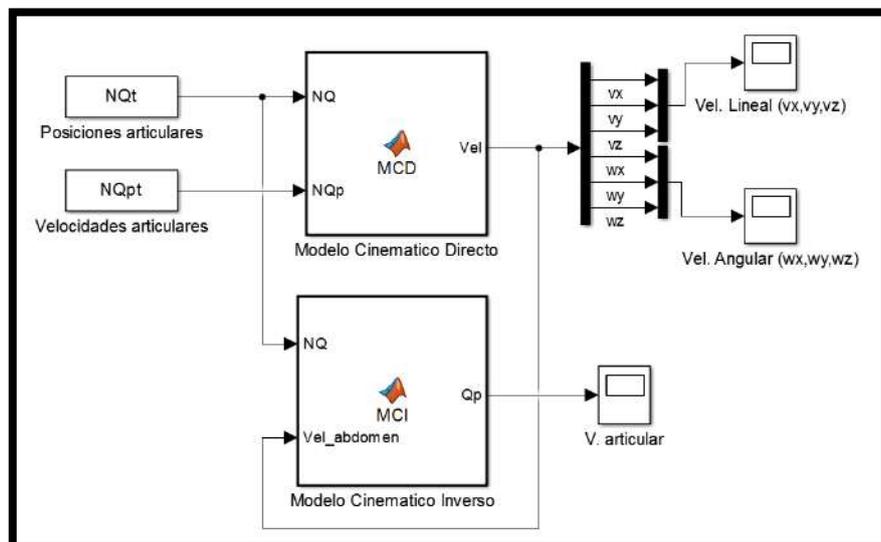


Figura 4.22 Prueba Modelo Cinemático

Para la simulación se han tomado las señales construidas anteriormente en series de tiempo, de manera tal que la trayectoria esférica que consta de 340 puntos sea recorrida en

1 segundo cada intervalo, es decir, se tendrá una velocidad promedio entre punto y punto de 1.115 cm/s. Esta consideración es válida debido a las bajas velocidades de movimiento que se requerirá dentro del abdomen.

Una vez determinadas las posiciones y velocidades articulares, utilizaremos el *modelo cinemático directo* para calcular las velocidades lineales (v_x , v_y , v_z) y angulares del laparoscopio (w_x , w_y , w_z).

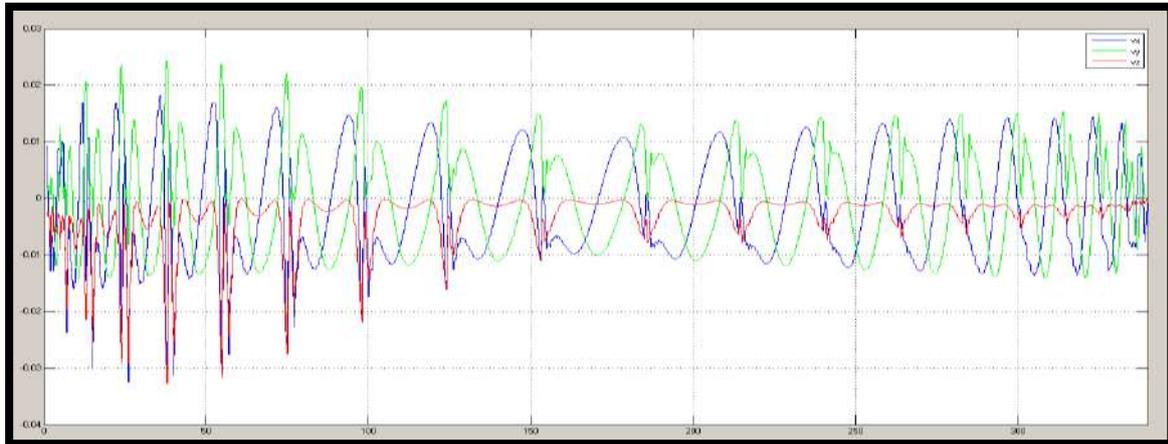


Figura 4.23 Velocidades lineales del laparoscopio [m/s]

En la Figura 4.23 se puede apreciar que el cambio de velocidad se produce más seguido en la parte inicial y final de la trayectoria, esto se debe a que los puntos construidos se encuentran más cercanos que en la parte media de la esfera. Además, la velocidad en 'z' es negativa ya que el movimiento siempre es hacia abajo y existen picos cercanos a 1.8 cm/s en el eje 'x' y de 2.4 cm/s en el eje 'y'. Las inflexiones que se muestran en los 3 ejes se dan cuando el robot ha terminado de seguir una circunferencia y pasa a la siguiente.

En la Figura 4.24 se muestra las velocidades angulares del laparoscopio medidas desde la base del robot.

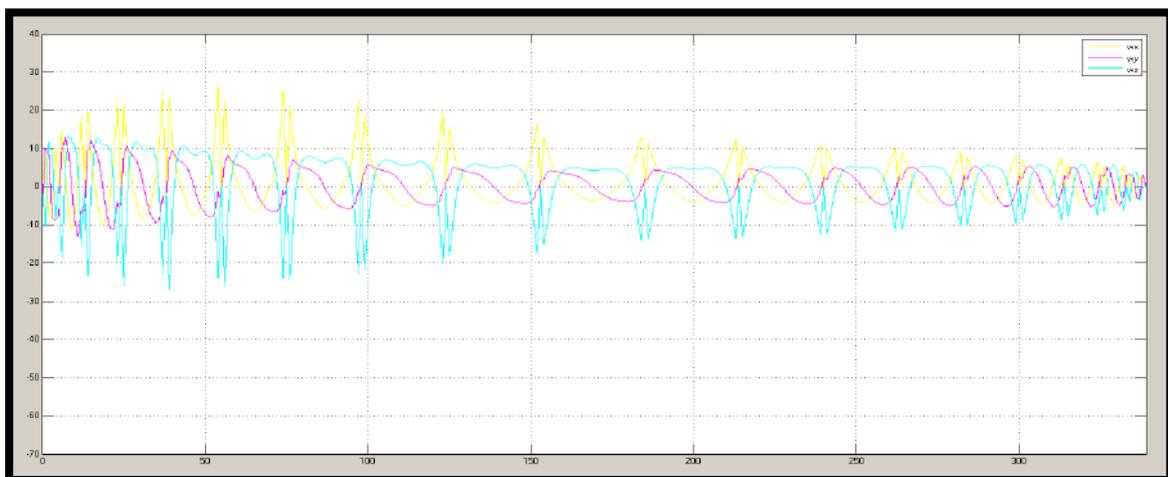


Figura 4.24 Velocidades angulares del laparoscopio [°/s]

Por otro lado, en la Figura 4.22 también se muestra el *modelo cinemático inverso*, en el cual se introducen estas velocidades calculadas (lineales y angulares), que junto con las posiciones articulares (que permiten encontrar las matrices de transformación homogénea) calculan las velocidades articulares.

Ahora, si bien uno de las principales utilidades del Jacobiano es que nos permite encontrar configuraciones singulares en aquellas en las que su determinante es cero, para el caso del interior del abdomen se ha dimensionado el brazo de forma tal que permita cubrir todo su espacio de trabajo.

4.6 MODELAMIENTO DINÁMICO IMEBOT

Hasta ahora, sólo se ha relacionado las posiciones y las velocidades dentro del abdomen, con sus correspondientes valores en el espacio articular. Sin embargo, no se han tomado en cuenta las fuerzas que las ocasionan. Para esto, debemos utilizar modelos dinámicos, que permiten relacionar las fuerzas o torques ejercidas por los actuadores y el movimiento producido en las articulaciones.

El desarrollo de un estudio dinámico involucra definir y extraer las características dinámicas del asistente robótico. Por esta razón, ha sido necesario la creación de un diseño mecánico en 3D, con los materiales y dimensiones físicas reales, que nos permita obtener numéricamente estos valores. De él conseguiremos:

- Masas
- Centros de gravedad: respecto a su sistema de coordenadas D-H
- Tensores de inercia

El modelo dinámico es imprescindible para conseguir ^[31]

- Simulación del movimiento del robot
- Diseño y evaluación de la estructura mecánica del robot
- Dimensionamiento de los actuadores
- Diseño y evaluación del control dinámico del robot.

Se debe tener presente que los robots son sistemas mecánicos muy complejos, de naturaleza no lineal, multivariable y fuerte acoplamiento. El modelo dinámico del robot manipulador permite explicar todos los fenómenos físicos que se encuentran en su estructura mecánica, tales como efectos inerciales, fuerzas centrípetas y de Coriolis, par gravitacional y fricción. ^[34]

De la misma forma que con los modelos geométricos y cinemáticos, para el modelamiento dinámico podemos calcular el *modelo dinámico inverso* (dadas las posiciones, velocidades y aceleraciones podemos calcular los pares necesarios para alcanzar estos valores) y el *modelo dinámico directo* (dados unos pares aplicados a las articulaciones se puede conocer su movimiento).

Existen varios métodos de modelado dinámico inverso como el de Newton – Euler, que es un método iterativo basado en el *balance de fuerzas*, y el de Euler – Lagrange basado en un *balance de energía* [35].

En el método de Lagrange se ofrece una solución con ecuaciones en forma cerrada, en las que permite distinguir fácilmente la estructura completa y las matrices que lo conforman (inercias, coriolis y gravedad), mientras que en el método Newton – Euler las ecuaciones son abiertas ya que se resuelve de manera iterativa y se llega directamente a los valores de las fuerzas y torques sin conocer los valores de las matrices [36].

Además también existen librerías en Matlab que permiten su cálculo, como lo es el ‘Toolbox de Robótica de Peter Corke’. En este trabajo se han realizado las 3 formas de calcular el modelo dinámico.

4.6.1 Parámetros inerciales

La extracción de parámetros inerciales se hizo a partir del diseño de IMEbot en Solidworks, que se presentó en el apartado 4.2. Este modelo CAD nos permite obtener las masas, posiciones del centro de gravedad y los tensores de inercia.

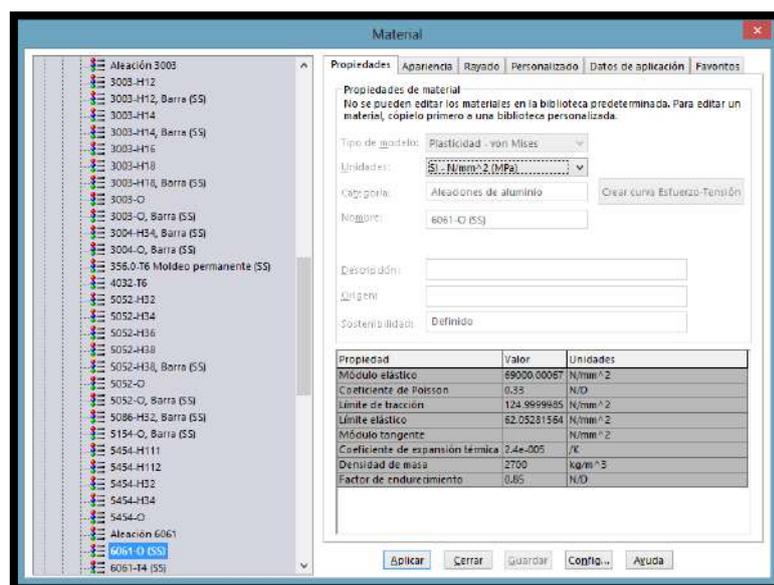


Figura 4.25 Asignación de material en la pieza

El primer paso es editar el material de las piezas en SOLIDWORKS, como se muestra en la Figura 4.25, ya que el software utiliza la densidad correspondiente y el volumen de la pieza para el cálculo de su masa.

Luego, es necesario asignar los sistemas de coordenadas de acuerdo a la convención Denavit – Hartenberg (Figura 4.26), ya que a partir de estos se miden los centros de gravedad y se obtiene los tensores de inercia (Figura 4.27).

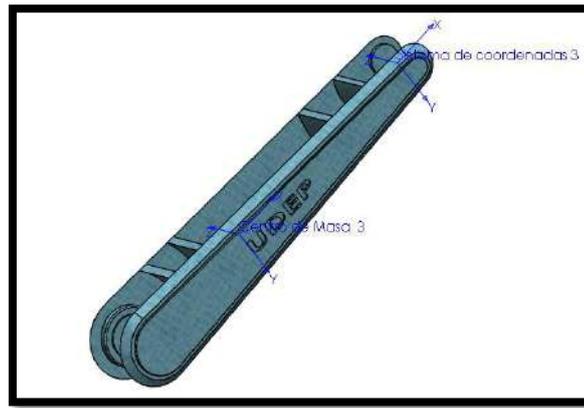


Figura 4.26 Asignación de sistemas de coordenadas D-H

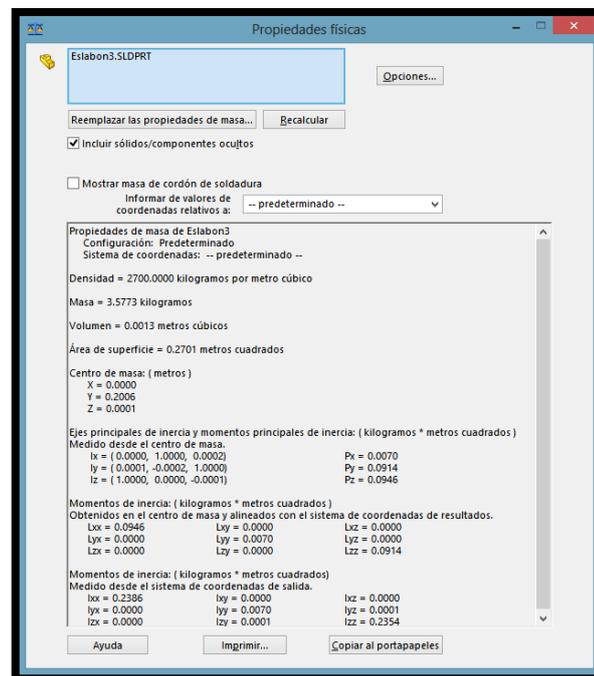


Figura 4.27 Parámetros inerciales

| | M_i | V_i | ρ_i |
|------------------|--------|-----------|----------|
| Eslabón 1 | 5.9664 | 0.0033 | 2700 |
| Eslabón 2 | 1.7791 | 0.0007 | 2700 |
| Eslabón 3 | 3.5773 | 0.0013 | 2700 |
| Eslabón 4 | 1.9849 | 0.0007 | 2700 |
| Eslabón 5 | 1.3959 | 0.0005 | 2700 |
| Eslabón 6 | 0.1200 | 0.0000444 | 2700 |
| Eslabón 7 | 0.7399 | 0.0000925 | 8000 |

Tabla 4.3 Parámetros inerciales: masa y densidad

Para extraer los parámetros dinámicos mostrados en la Tabla 4.3 y la Tabla 4.4 se debe ir a la pestaña *Calcular >> Propiedades Físicas*. Una vez dentro de ésta, se pueden cambiar las *Opciones* para seleccionar el sistema de coordenadas creado previamente y se ajustan las unidades de salida para las masas y tensores de inercia.

| | r_i | | | I_i | | |
|------------------|---------|---------|---------|---------|---------|------------|
| | rx_i | ry_i | rz_i | Ixx_i | Iyy_i | Izz_i |
| <i>Eslabón 1</i> | 0 | 0 | -0.1603 | 0.4105 | 0.4105 | 0.0274 |
| <i>Eslabón 2</i> | 0 | 0.0473 | 0 | 0.0082 | 0.0021 | 0.0086 |
| <i>Eslabón 3</i> | -0.2994 | 0 | 0 | 0.0097 | 0.7038 | 0.7010 |
| <i>Eslabón 4</i> | 0 | 0 | 0.0377 | 0.0075 | 0.0076 | 0.0016 |
| <i>Eslabón 5</i> | 0 | -0.1179 | 0 | 0.0306 | 0.0006 | 0.0016 |
| <i>Eslabón 6</i> | 0 | 0 | 0.0273 | 0.0002 | 0.0002 | 0.00001166 |
| <i>Eslabón 7</i> | 0 | 0 | -0.3116 | 0.0749 | 0.0749 | 0.0001 |

Tabla 4.4 Parámetros inerciales: centro de gravedad y tensor de inercias

Donde:

- M_i : masa del eslabón i [Kg]
- V_i : volumen del eslabón i [m³]
- ρ_i : densidad del eslabón i [m³/Kg]
- r_i : distancia del centro de gravedad respecto a su sistema de coordenadas DH [m].
- I_i : momentos de inercia respecto a su sistema de coordenadas DH. [Kg*m²]

4.6.2 Formulación de Euler-Lagrange ^[37]

El método de Lagrange está basado en el balance de energía del sistema (energía cinética y potencial), los cuales utilizan las leyes de Newton y la mecánica lagrangiana. Éste utiliza la matriz de transformación Denavit – Hartenberg para la representación espacial entre dos sistemas de coordenadas consecutivos.

Uno de los principales beneficios de emplear este método es que conduce a unas ecuaciones finales bien estructuradas donde aparecen de manera clara los diversos pares y fuerzas que intervienen en el movimiento (inercia, Coriolis, gravedad) ^[31].

La ecuación Euler - Lagrange:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = \tau_i \quad (4.78)$$

El lagrangiano está definido como:

$$L = K - P \quad (4.79)$$

Donde:

- L : Función Lagrangiana
- K : Energía cinética
- P : Energía potencial
- \mathbf{q}_i : coordenadas generalizadas del robot
- $\dot{\mathbf{q}}_i$: primera derivada de las coordenadas generalizadas,
- τ_i : fuerza o torque aplicado al sistema en la articulación i .

4.6.2.1 Velocidades articulares de un robot manipulador

La formulación lagrangiana necesita conocer la energía cinética del sistema físico, para lo cual se requiere saber la velocidad de cada articulación.

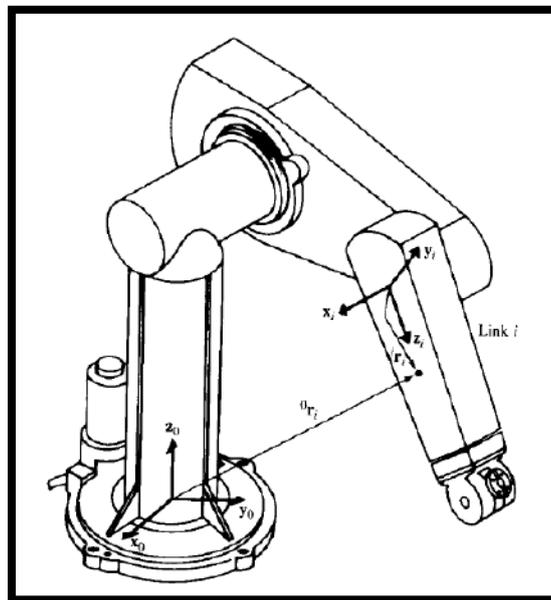


Figura 4.28 Ubicación de un punto arbitrario 0r_i sobre el eslabón i . [37]

Sea ${}^i r_i$ un vector fijo expresado en el sistema de coordenadas i .

$${}^i \bar{r}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \\ 1 \end{pmatrix}$$

Sea ${}^0 r_i$ el mismo punto que ${}^i r_i$, pero con respecto a la base, entonces

$${}^0 \bar{r}_i = {}^0 A_i * {}^i \bar{r}_i$$

Donde ${}^0 A_i = {}^0 A_1 * {}^1 A_2 * \dots * {}^{i-1} A_i$

Si la articulación es revoluta, la matriz de transformación será:

$${}^{i-1}\mathbf{A}_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & \alpha_i \cos \theta_i \\ \cos \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & \alpha_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Si la articulación es prismática, la matriz de transformación será:

$${}^{i-1}\mathbf{A}_i = \begin{bmatrix} \cos \theta_i & -\cos \alpha_i \sin \theta_i & \sin \alpha_i \sin \theta_i & 0 \\ \cos \theta_i & \cos \alpha_i \cos \theta_i & -\sin \alpha_i \cos \theta_i & 0 \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Dado que ${}^i\mathbf{r}_i$ es fijo con respecto a su sistema de coordenadas, no tendrá velocidad. Por lo tanto, la velocidad de ${}^i\mathbf{r}_i$ con respecto a la base será:

$$\begin{aligned} {}^0\bar{\mathbf{v}}_i &= \bar{\mathbf{v}}_i = \frac{d}{dt} ({}^0\bar{\mathbf{r}}_i) = \frac{d}{dt} ({}^0\mathbf{A}_i {}^i\bar{\mathbf{r}}_i) \\ {}^0\bar{\mathbf{v}}_i &= ({}^0\dot{\mathbf{A}}_1 {}^1\mathbf{A}_2 \dots {}^{i-1}\mathbf{A}_i {}^i\bar{\mathbf{r}}_i) + ({}^0\mathbf{A}_1 {}^1\dot{\mathbf{A}}_2 \dots {}^{i-1}\mathbf{A}_i {}^i\bar{\mathbf{r}}_i) + \dots \\ &\quad + ({}^0\mathbf{A}_1 {}^1\mathbf{A}_2 \dots {}^{i-1}\dot{\mathbf{A}}_i {}^i\bar{\mathbf{r}}_i) + ({}^0\mathbf{A}_1 {}^1\mathbf{A}_2 \dots {}^{i-1}\mathbf{A}_i {}^i\dot{\bar{\mathbf{r}}}_i) \\ {}^0\bar{\mathbf{v}}_i &= \left(\sum_{j=1}^i \frac{\partial {}^0\mathbf{A}_i}{\partial \mathbf{q}_j} \dot{\mathbf{q}}_j \right) {}^i\bar{\mathbf{r}}_i \end{aligned} \quad (4.80)$$

Ahora, la derivada parcial de ${}^0\mathbf{A}_i$ con respecto a \mathbf{q}_j se puede calcular fácilmente con ayuda de una matriz \mathbf{Q}_i , la cual está definida como:

$$\mathbf{Q}_{i_revoluta} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{Q}_{i_prismatico} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Entonces:

$$\frac{\partial {}^{i-1}\mathbf{A}_i}{\partial \mathbf{q}_i} = \mathbf{Q}_i * {}^{i-1}\mathbf{A}_i \quad (4.81)$$

Por lo que, para $i=1,2,\dots,n$

$$\frac{\partial {}^0\mathbf{A}_i}{\partial \mathbf{q}_j} = \begin{cases} {}^0\mathbf{A}_i {}^1\mathbf{A}_2 \dots {}^{j-2}\mathbf{A}_{j-1} \mathbf{Q}_j {}^{j-1}\mathbf{A}_j \dots {}^{i-1}\mathbf{A}_i & \text{para } j \leq i \\ 0 & \text{para } j > i \end{cases}$$

También podemos escribir lo anterior como:

$$\mathbf{U}_{ij} = \frac{\partial {}^0\mathbf{A}_i}{\partial \mathbf{q}_j} = \begin{cases} {}^0\mathbf{A}_{j-1} * \mathbf{Q}_j * {}^{j-1}\mathbf{A}_i & \text{para } j \leq i \\ 0 & \text{para } j > i \end{cases} \quad (4.82)$$

Entonces, reemplazando en la velocidad, tendremos que:

$$\bar{\mathbf{v}}_i = \left(\sum_{j=1}^i \mathbf{U}_{ij} \dot{\mathbf{q}}_j \right) {}^i\bar{\mathbf{r}}_i \quad (4.83)$$

Podemos apreciar que la ventaja de usar las matrices \mathbf{Q}_i es que nos permite seguir usando las matrices de transformación homogénea ${}^{i-1}\mathbf{A}_i$.

Ahora, se necesita encontrar los efectos de la interacción entre articulaciones, es decir los efectos que tendrán las articulaciones j y k sobre todos los puntos ubicados en el eslabón i .

$$\frac{\partial \mathbf{U}_{ij}}{\partial \mathbf{q}_k} = \mathbf{U}_{ijk} = \begin{cases} {}^0\mathbf{A}_{j-1} * \mathbf{Q}_j * {}^{j-1}\mathbf{A}_{k-1} * \mathbf{Q}_k * {}^{k-1}\mathbf{A}_i & \text{para } i \geq k \geq j \\ {}^0\mathbf{A}_{k-1} * \mathbf{Q}_k * {}^{k-1}\mathbf{A}_{j-1} * \mathbf{Q}_j * {}^{j-1}\mathbf{A}_i & \text{para } i \geq j \geq k \\ 0 & \text{para } i < j \text{ ó } i < k \end{cases} \quad (4.84)$$

4.6.2.2 Energía Cinética de un Robot Manipulador

Una vez calculadas las velocidades de cada articulación, podremos calcular la energía cinética del eslabón i . Sea K_i la energía cinética de cada eslabón y dK_i la energía cinética de una partícula con una masa diferencial dm , entonces:

$$\begin{aligned} dK_i &= \frac{1}{2} (\dot{x}_i^2 + \dot{y}_i^2 + \dot{z}_i^2) dm \\ dK_i &= \frac{1}{2} \text{traza} (\bar{\mathbf{v}}_i \bar{\mathbf{v}}_i^T) dm = \frac{1}{2} \text{Tr} (\bar{\mathbf{v}}_i \bar{\mathbf{v}}_i^T) dm \end{aligned} \quad (4.85)$$

Reemplazando $\bar{\mathbf{v}}_i$ en la ecuación (4.85)

$$\begin{aligned} dK_i &= \frac{1}{2} \text{Tr} \left(\sum_{p=1}^i \mathbf{U}_{ip} \dot{\mathbf{q}}_p {}^i\bar{\mathbf{r}}_i * \left(\sum_{r=1}^i \mathbf{U}_{ir} \dot{\mathbf{q}}_r {}^i\bar{\mathbf{r}}_i \right)^T \right) dm \\ dK_i &= \frac{1}{2} \text{Tr} \left(\sum_{p=1}^i \sum_{r=1}^i \mathbf{U}_{ip} {}^i\bar{\mathbf{r}}_i {}^i\bar{\mathbf{r}}_i^T \mathbf{U}_{ir}^T \dot{\mathbf{q}}_p \dot{\mathbf{q}}_r \right) dm \\ dK_i &= \frac{1}{2} \text{Tr} \left(\sum_{p=1}^i \sum_{r=1}^i \mathbf{U}_{ip} ({}^i\bar{\mathbf{r}}_i dm {}^i\bar{\mathbf{r}}_i^T) \mathbf{U}_{ir}^T \dot{\mathbf{q}}_p \dot{\mathbf{q}}_r \right) \end{aligned} \quad (4.86)$$

Integrando para obtener la energía cinética correspondiente al eslabón i :

$$K_i = \frac{1}{2} \text{Tr} \left(\sum_{p=1}^i \sum_{r=1}^i \mathbf{U}_{ip} \left(\int {}^i \bar{\mathbf{r}}_i {}^i \bar{\mathbf{r}}_i^T dm \right) \mathbf{U}_{ir}^T \dot{\mathbf{q}}_p \dot{\mathbf{q}}_r \right) \quad (4.87)$$

Llamaremos J_i (matriz de pseudo-inercias) al término entre paréntesis, es decir:

$$J_i = \left(\int {}^i \bar{\mathbf{r}}_i {}^i \bar{\mathbf{r}}_i^T dm \right) = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (4.88)$$

Donde ${}^i \bar{\mathbf{r}}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i)^T$ es el vector de centro de gravedad medido desde su sistema D-H.

Por lo tanto, la energía total del asistente robótico será una magnitud escalar dependiente de la posición y velocidad de las articulaciones:

$$K = \sum_{i=1}^n K_i = \frac{1}{2} \left(\sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i \text{Tr} (\mathbf{U}_{ip} J_i \mathbf{U}_{ir}^T) \dot{\mathbf{q}}_p \dot{\mathbf{q}}_r \right) \quad (4.89)$$

Además J_i depende de la distribución de la masa del eslabón i y no de su posición ni de su velocidad de movimiento, así que sólo será necesario calcularlo una vez.

4.6.2.3 Energía Potencial de un robot Manipulador

Sea P_i la energía potencial de cada eslabón:

$$P_i = -m_i \bar{\mathbf{g}}^0 \bar{\mathbf{r}}_i = -m_i \bar{\mathbf{g}} ({}^0 \mathbf{A}_i * {}^i \bar{\mathbf{r}}_i)$$

La energía potencial total será:

$$P_i = \sum_{i=1}^n P_i = \sum_{i=1}^n (-m_i \bar{\mathbf{g}} ({}^0 \mathbf{A}_i * {}^i \bar{\mathbf{r}}_i)) \quad (4.90)$$

Donde $\bar{\mathbf{g}} = (g_x, g_y, g_z, 0)$ es el vector de gravedad expresado en el sistema de coordenadas base.

4.6.2.4 Ecuaciones de movimiento de un manipulador

Reemplazamos las ecuaciones (4.89) y (4.90) en la función lagrangiana (4.79):

$$L = \frac{1}{2} \left(\sum_{i=1}^n \sum_{p=1}^i \sum_{r=1}^i \text{Tr} (\mathbf{U}_{ip} J_i \mathbf{U}_{ir}^T) \dot{\mathbf{q}}_p \dot{\mathbf{q}}_r \right) + \sum_{i=1}^n (m_i \bar{\mathbf{g}} ({}^0\mathbf{A}_i * {}^i\bar{\mathbf{r}}_i)) \quad (4.91)$$

Aplicamos la formulación Euler-Lagrange (4.78) para encontrar los torques:

$$\tau_i = \sum_{j=1}^n \sum_{k=1}^j \text{Tr} (\mathbf{U}_{jk} J_j \mathbf{U}_{ji}^T) \ddot{\mathbf{q}}_k + \sum_{j=1}^n \sum_{k=1}^j \sum_{m=1}^j \text{Tr} (\mathbf{U}_{jkm} J_j \mathbf{U}_{ji}^T) \dot{\mathbf{q}}_k \dot{\mathbf{q}}_m - \sum_{j=i}^n (m_j \bar{\mathbf{g}} \mathbf{U}_{ji} {}^j\bar{\mathbf{r}}_j) \quad (4.92)$$

Para $i, k, m = 1, 2, \dots, n$. Donde n son el número de articulaciones.

La ecuación (4.92) se puede expresar en forma resumida como:

$$\tau_i = \sum_{k=1}^n \mathbf{D}_{ik} \ddot{\mathbf{q}}_k + \sum_{k=1}^n \sum_{m=1}^n \mathbf{h}_{ikm} \dot{\mathbf{q}}_k \dot{\mathbf{q}}_m - \mathbf{C}_i \quad (4.93)$$

O en forma matricial como:

$$\boldsymbol{\tau}(t) = \mathbf{D}(q(t)) \ddot{\mathbf{q}}(t) + \mathbf{H}(q(t), \dot{\mathbf{q}}(t)) + \mathbf{C}(q(t)) \quad (4.94)$$

Donde:

- $\mathbf{D}(q(t))$: es la matriz de inercias, de dimensión $n \times n$, cuyos elementos son:

$$\mathbf{D}_{ik} = \sum_{j=\max(i,k)}^n \text{Tr} (\mathbf{U}_{jk} J_j \mathbf{U}_{ji}^T) \quad (4.95)$$

Cuando $i = k$, \mathbf{D}_{ii} es llamada *inercia efectiva* y relaciona el torque que se produce i a causa de una aceleración en i , mientras que si $i \neq k$, \mathbf{D}_{ik} es llamada *inercia acoplada* y relaciona el torque que se produce i a causa de una aceleración en k .

- $\mathbf{H}(q(t))$: es la matriz de fuerzas centrípeta y de Coriolis, de dimensión $n \times 1$:

$$h_i = \sum_{k=1}^n \sum_{m=1}^n h_{ikm} \dot{\mathbf{q}}_k \dot{\mathbf{q}}_m \quad (4.96)$$

$$h_{ikm} = \sum_{j=\max(i,k,m)}^n \text{Tr} (\mathbf{U}_{jkm} J_j \mathbf{U}_{ji}^T) \quad (4.97)$$

Cuando $k = m$, \mathbf{h}_{ikm} es la *fuerza centrífuga* y relaciona el torque que se produce i a causa de la velocidad en k , mientras que si $k \neq m$, \mathbf{h}_{ikm} es la *fuerza de Coriolis* y relaciona el torque que se produce i a causa de las velocidades en k y m .

- $\mathbf{C}(q(t))$: es la matriz de fuerzas de gravedad, de dimensión $n \times 1$, cuyos elementos son:

$$C_i = \sum_{j=i}^n \left(-m_j \bar{g} U_{ji} {}^j \bar{r}_j \right) \quad (4.98)$$

Las ecuaciones de movimiento Euler – Lagrange presentan la ventaja de tener su estructura formada por matrices, lo cual es deseable para poder generar lazos de control y se utilizan sus coeficientes para minimizar los efectos no lineales. Todas estas ecuaciones se han implementado en Matlab en la función “MDI_LG.m” y se muestran en el anexo 9.2.6.1.

4.6.3 Formulación de Newton – Euler ^[37]

El método de Newton-Euler aparece como alternativa para derivar de una manera más eficiente las ecuaciones del movimiento y así poder diseñar un sistema de control en tiempo real. Esto debido a que la formulación Lagrangiana usa las matrices de transformación homogénea 4x4.

Las ecuaciones dinámicas resultantes son un conjunto de ecuaciones recursivas hacia delante y hacia atrás y pueden ser aplicadas secuencialmente a los eslabones del robot. Las ecuaciones hacia delante nos dan la información cinemática de cada eslabón, es decir, las velocidades lineales, velocidades angulares, aceleraciones angulares y aceleraciones lineales del centro de gravedad. Las ecuaciones hacia atrás nos dan la información sobre las fuerzas y torques aplicadas en cada eslabón y como se propagan desde el efector final hasta sistema de coordenadas base.

4.6.3.1 Cinemática de los eslabones

Lo primero que se debe hacer es encontrar las ecuaciones cinemáticas que relacionen el movimiento (rotación y traslación) de los eslabones de un robot con su respectivo sistema base. Para ello debemos tener en cuenta las siguientes relaciones matemáticas:

Sistema de Coordenadas Giratorio: Sea XYZ un sistema de referencia fijo y X*Y*Z* un sistema que gira respecto al primero, entonces podemos expresar un punto r como:

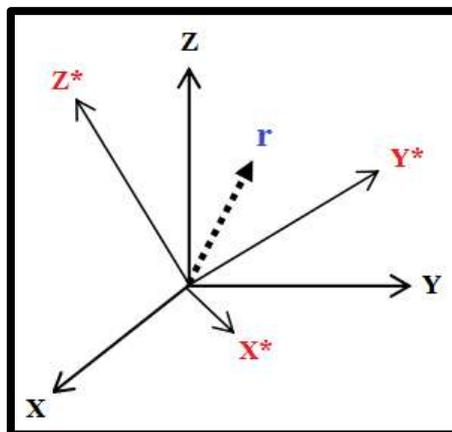


Figura 4.29 Sistema de coordenadas giratorio

$$\bar{\mathbf{r}} = x\bar{i} + y\bar{j} + z\bar{k} \quad (4.99)$$

$$\bar{\mathbf{r}} = x^*\bar{i}^* + y^*\bar{j}^* + z^*\bar{k}^* \quad (4.100)$$

Sus derivadas con respecto al tiempo para sistemas de coordenadas giratorios serán:

$$\frac{d\bar{\mathbf{r}}}{dt} = \frac{d^*\bar{\mathbf{r}}}{dt} + \bar{\boldsymbol{\omega}} \times \bar{\mathbf{r}} \quad (4.101)$$

$$\frac{d^2\bar{\mathbf{r}}}{dt^2} = \frac{d^{*2}\bar{\mathbf{r}}}{dt^2} + 2\bar{\boldsymbol{\omega}} \times \frac{d^*\bar{\mathbf{r}}}{dt} + \bar{\boldsymbol{\omega}} \times (\bar{\boldsymbol{\omega}} \times \bar{\mathbf{r}}) + \frac{d\bar{\boldsymbol{\omega}}}{dt} \times \bar{\mathbf{r}} \quad (4.102)$$

En la ecuación (4.102) podemos apreciar la aceleración relativa al sistema de coordenadas $X^*Y^*Z^*$, la aceleración de Coriolis y la aceleración centrípeta.

Sistema de Coordenadas en Movimiento: El sistema $X^*Y^*Z^*$ se está trasladando y rotando respecto a XYZ . Sea p una partícula que puede ser representado en ambos sistemas:

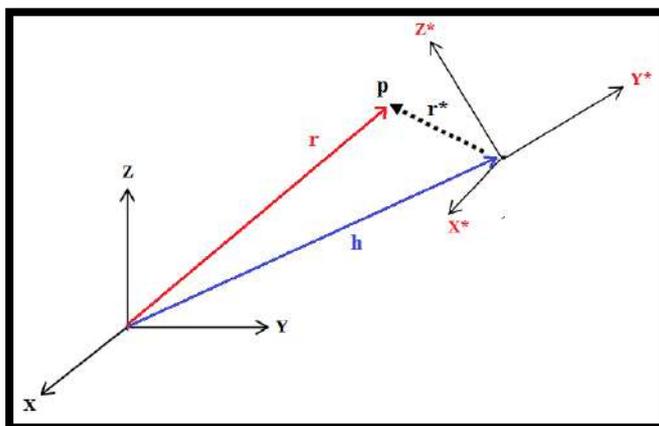


Figura 4.30 Sistema de coordenadas en rotación y traslación

$$\bar{\mathbf{r}} = \bar{\mathbf{r}}^* + \bar{\mathbf{h}} \quad (4.103)$$

$$\bar{\mathbf{v}}(t) = \bar{\mathbf{v}}^* + \bar{\mathbf{v}}_h = \frac{d\bar{\mathbf{r}}^*}{dt} + \frac{d\bar{\mathbf{h}}}{dt} \quad (4.104)$$

$$\bar{\mathbf{a}}(t) = \bar{\mathbf{a}}^* + \bar{\mathbf{a}}_h = \frac{d^2\bar{\mathbf{r}}^*}{dt^2} + \frac{d^2\bar{\mathbf{h}}}{dt^2} \quad (4.105)$$

Usando la ecuación (4.101) en (4.104)

$$\bar{\mathbf{v}}(t) = \left(\frac{d^*\bar{\mathbf{r}}^*}{dt} + \bar{\boldsymbol{\omega}} \times \bar{\mathbf{r}}^* \right) + \frac{d\bar{\mathbf{h}}}{dt} \quad (4.106)$$

Usando la ecuación (4.102) en (4.105)

$$\bar{\mathbf{a}}(t) = \left(\frac{d^{*2}\bar{\mathbf{r}}^*}{dt^2} + 2\bar{\boldsymbol{\omega}} \times \frac{d^*\bar{\mathbf{r}}^*}{dt} + \bar{\boldsymbol{\omega}} \times (\bar{\boldsymbol{\omega}} \times \bar{\mathbf{r}}^*) + \frac{d\bar{\boldsymbol{\omega}}}{dt} \times \bar{\mathbf{r}}^* \right) + \frac{d^2\bar{\mathbf{h}}}{dt^2} \quad (4.107)$$

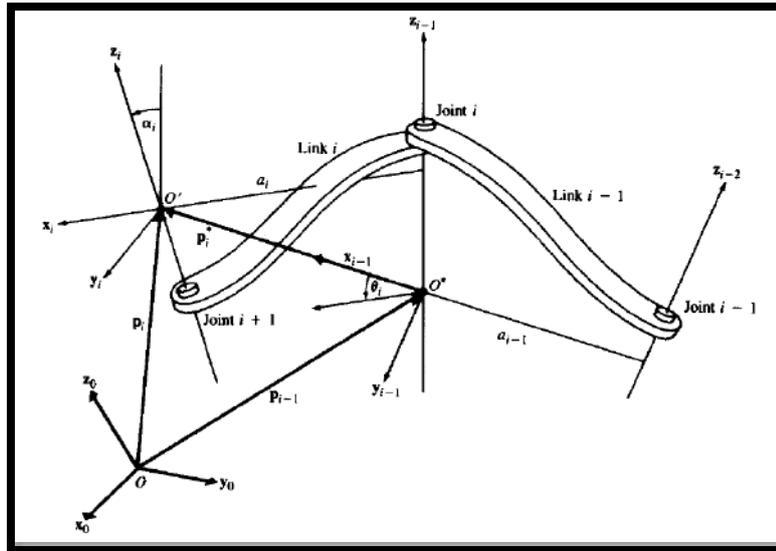


Figura 4.31 Relación entre los sistemas O, O^* y O' [37]

Utilizando la ecuación (4.106) y cambiando los valores de acuerdo a la Figura 4.31, tenemos:

$$\bar{v}_i(t) = \left(\frac{d^* \bar{p}_i}{dt} + \bar{\omega}_{i-1} \times \bar{p}_i^* \right) + \bar{v}_{i-1} \quad (4.108)$$

$$\bar{\omega}_i = \bar{\omega}_{i-1} + \bar{\omega}_i^* \quad (4.109)$$

Utilizando la ecuación (4.107) y cambiando los valores de acuerdo a la Figura 4.31, tenemos:

$$\bar{v}_i(t) = \left(\frac{d^{*2} \bar{p}_i}{dt^2} + 2\bar{\omega}_{i-1} \times \frac{d^* \bar{p}_i}{dt} + \bar{\omega}_{i-1} \times (\bar{\omega}_{i-1} \times \bar{p}_i^*) + \dot{\bar{\omega}}_{i-1} \times \bar{p}_i^* \right) + \bar{v}_{i-1} \quad (4.110)$$

$$\bar{\omega}_i = \bar{\omega}_{i-1} + \bar{\omega}_i^* \quad (4.111)$$

Si reemplazamos al vector $\bar{\omega}_i^*$ utilizando (4.101), entonces (4.111) quedaría como:

$$\bar{\omega}_i = \bar{\omega}_{i-1} + \frac{d^* \bar{\omega}_i^*}{dt} + \bar{\omega}_{i-1} \times \bar{\omega}_i^* \quad (4.112)$$

Debido a la convención Denavit-Hartenberg para la asignación de los sistemas de coordenadas, se puede notar que el movimiento entre el sistema $\{S_{i-1}\}$ y el $\{S_i\}$ será en siempre en el eje z_{i-1} . Si el eslabón i es prismático en el sistema $\{S_{i-1}\}$, se producirá un movimiento en la dirección z_{i-1} con una *velocidad lineal* igual a \dot{q}_i . Si el eslabón i es rotacional en el sistema $\{S_{i-1}\}$, se producirá un movimiento en la dirección z_{i-1} con una *velocidad angular* igual a \dot{q}_i . Por lo tanto:

$$\bar{\omega}_i^* = \begin{cases} z_{i-1} \dot{q}_i, & \text{si } i \text{ es rotacional} \\ 0, & \text{si } i \text{ es prismático} \end{cases} \quad (4.113)$$

$$\frac{d^* \bar{\omega}_i^*}{dt} = \begin{cases} z_{i-1} \ddot{q}_i, & \text{si } i \text{ es rotacional} \\ 0, & \text{si } i \text{ es prismático} \end{cases} \quad (4.114)$$

Reemplazamos (4.113) y (4.114) en (4.109) y (4.112) respectivamente, entonces:

$$\bar{\omega}_i = \begin{cases} \bar{\omega}_{i-1} + \bar{z}_{i-1}\dot{q}_i, & \text{si } i \text{ es rotacional} \\ \bar{\omega}_{i-1}, & \text{si } i \text{ es prismático} \end{cases} \quad (4.115)$$

$$\bar{\dot{\omega}}_i = \begin{cases} \bar{\dot{\omega}}_{i-1} + \bar{z}_{i-1}\ddot{q}_i + \bar{\omega}_{i-1} \times (\bar{z}_{i-1}\dot{q}_i), & \text{si } i \text{ es rotacional} \\ \bar{\dot{\omega}}_{i-1}, & \text{si } i \text{ es prismático} \end{cases} \quad (4.116)$$

Ahora, sabiendo que la velocidad lineal de un vector \bar{s} está definido que está girando con una velocidad angular $\bar{\omega}$ está dado por:

$$\bar{v}_s = \bar{\omega} \times \bar{s}_i$$

Entonces la velocidad y aceleración lineal del sistema $\{S_i\}$ con respecto a $\{S_{i-1}\}$, es decir, del punto \bar{p}_i^* con respecto al sistema $\{X_{i-1}Y_{i-1}Z_{i-1}\}$ es:

$$\frac{d^*\bar{p}_i^*}{dt} = \begin{cases} \bar{\omega}_i^* \times \bar{p}_i^*, & \text{si } i \text{ es rotacional} \\ \bar{z}_{i-1}\dot{q}_i, & \text{si } i \text{ es prismático} \end{cases} \quad (4.117)$$

$$\frac{d^{*2}\bar{p}_i^*}{dt^2} = \begin{cases} \frac{d^*\bar{\omega}_i^*}{dt} \times \bar{p}_i^* + \bar{\omega}_i^* \times (\bar{\omega}_i^* \times \bar{p}_i^*), & \text{si } i \text{ es rotacional} \\ \bar{z}_{i-1}\ddot{q}_i, & \text{si } i \text{ es prismático} \end{cases} \quad (4.118)$$

Reemplazando (4.117) y (4.118) en (4.108) y (4.110) respectivamente y tomando en cuenta (4.109) y (4.111) tendremos:

$$\frac{d^*\bar{p}_i^*}{dt} = \begin{cases} \bar{\omega}_i^* \times \bar{p}_i^*, & \text{si } i \text{ es rotacional} \\ \bar{z}_{i-1}\dot{q}_i, & \text{si } i \text{ es prismático} \end{cases} \quad (4.119)$$

$$\frac{d^{*2}\bar{p}_i^*}{dt^2} = \begin{cases} \frac{d^*\bar{\omega}_i^*}{dt} \times \bar{p}_i^* + \bar{\omega}_i^* \times (\bar{\omega}_i^* \times \bar{p}_i^*), & \text{si } i \text{ es rotacional} \\ \bar{z}_{i-1}\ddot{q}_i, & \text{si } i \text{ es prismático} \end{cases} \quad (4.120)$$

$$\bar{v}_i(t) = \begin{cases} \bar{\omega}_i \times \bar{p}_i^* + \bar{v}_{i-1}, & \text{si } i \text{ es rotacional} \\ \bar{z}_{i-1}\dot{q}_i + \bar{\omega}_{i-1} \times \bar{p}_i^* + \bar{v}_{i-1}, & \text{si } i \text{ es prismático} \end{cases} + \quad (4.121)$$

$$\bar{\dot{v}}_i = \begin{cases} \bar{\dot{\omega}}_i \times \bar{p}_i^* + \bar{\omega}_i \times (\bar{\omega}_i \times \bar{p}_i^*) + \bar{\dot{v}}_{i-1}, & \text{si } i \text{ es rotacional} \\ \bar{z}_{i-1}\ddot{q}_i + \bar{\dot{\omega}}_i \times \bar{p}_i^* + 2\bar{\omega}_i \times (\bar{z}_{i-1}\dot{q}_i) + \bar{\omega}_i \times (\bar{\omega}_i \times \bar{p}_i^*) + \bar{\dot{v}}_{i-1}, & \text{prism.} \end{cases} \quad (4.122)$$

4.6.3.2 Ecuaciones recursivas del movimiento del manipulador

Aplicaremos la segunda ley de Newton y el principio de d'Alembert en cada eslabón:

$$\mathbf{F}_i = \frac{dm_i \bar{v}_{ci}}{dt} = m_i \bar{a}_{ci} \quad (4.123)$$

$$\mathbf{N}_i = \frac{d(\mathbf{I}_i \boldsymbol{\omega}_i)}{dt} = \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i) \quad (4.124)$$

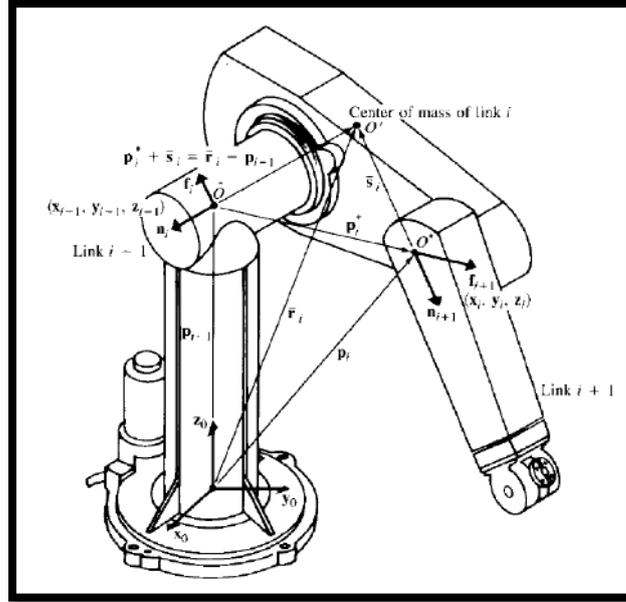


Figura 4.32 Fuerzas y momentos sobre eslabón i [37]

Si usamos las ecuaciones (4.121) y (4.122), las velocidades y aceleraciones lineales para los centros de masas:

$$\bar{\mathbf{v}}_{ci} = \bar{\boldsymbol{\omega}}_i \times \bar{\mathbf{s}}_i + \bar{\mathbf{v}}_i \quad (4.125)$$

$$\bar{\mathbf{a}}_{ci} = \dot{\bar{\boldsymbol{\omega}}}_i \times \bar{\mathbf{s}}_i + \bar{\boldsymbol{\omega}}_i \times (\bar{\boldsymbol{\omega}}_i \times \bar{\mathbf{s}}_i) + \bar{\mathbf{v}}_i \quad (4.126)$$

De la Figura 4.32 analizamos la fuerza total \mathbf{F}_i y el torque total \mathbf{N}_i que son aplicados al eslabón i . Éstos son resultado de la gravedad y la interacción con el eslabón anterior y posterior, entonces:

$$\mathbf{F}_i = \mathbf{f}_i - \mathbf{f}_{i+1} \quad (4.127)$$

$$\mathbf{N}_i = \mathbf{n}_i + (\bar{\mathbf{p}}_{i-1} - \bar{\mathbf{r}}_i) \times \mathbf{f}_i - \mathbf{n}_{i+1} - (\bar{\mathbf{p}}_i - \bar{\mathbf{r}}_i) \times \mathbf{f}_{i+1} \quad (4.128)$$

Despejamos las fuerzas y torques aplicadas solamente en la articulación i y sabiendo que $(\bar{\mathbf{r}}_i - \bar{\mathbf{p}}_{i-1}) = (\bar{\mathbf{p}}_i^* + \bar{\mathbf{s}}_i)$

$$\mathbf{f}_i = m_i \bar{\mathbf{a}}_{ci} + \mathbf{f}_{i+1} \quad (4.129)$$

$$\mathbf{n}_i = \mathbf{N}_i + (\bar{\mathbf{p}}_i^* + \bar{\mathbf{s}}_i) \times \mathbf{F}_i + \mathbf{n}_{i+1} + \bar{\mathbf{p}}_i^* \times \mathbf{f}_{i+1} \quad (4.130)$$

Donde:

- m_i : Masa del eslabón i .
- $\bar{\mathbf{s}}_i$: Posición del centro de gravedad del eslabón i respecto al sistema $\{S_i\}$
- $\bar{\mathbf{p}}_i^*$: Origen del sistema $\{S_i\}$ con respecto al anterior $\{S_{i-1}\}$.
- $\bar{\mathbf{v}}_i$: Aceleración lineal del sistema i
- $\bar{\mathbf{v}}_{ci}$: Velocidad lineal del centro de gravedad del eslabón i
- $\bar{\mathbf{a}}_{ci}$: Aceleración lineal del centro de gravedad del eslabón i .
- \mathbf{F}_i : Fuerza total externa ejercida sobre el eslabón i en el centro de gravedad.

- N_i : Torque total externo ejercido sobre el eslabón i en el centro de gravedad.
- I_i : Matriz de inercia del eslabón i con origen en el centro de gravedad respecto a la base.
- f_i : Fuerza ejercida sobre el eslabón i por el eslabón $i-1$ en el origen del sistema $\{S_{i-1}\}$ para soportar al eslabón i y los eslabones encima de él.
- n_i : Torque ejercido sobre el eslabón i por el eslabón $i-1$ en el origen del sistema $\{S_{i-1}\}$.

Resumiendo, el método de Newton – Euler para calcular las fuerzas y torques aplicadas en las articulaciones se resume en las siguientes ecuaciones:

Ecuaciones hacia adelante: $i = 1, 2, \dots, n$

$$\boldsymbol{\omega}_i = \begin{cases} \boldsymbol{\omega}_{i-1} + \bar{\mathbf{z}}_{i-1} \dot{q}_i, & \text{si } i \text{ es rotacional} \\ \boldsymbol{\omega}_{i-1}, & \text{si } i \text{ es prismático} \end{cases}$$

$$\dot{\boldsymbol{\omega}}_i = \begin{cases} \dot{\boldsymbol{\omega}}_{i-1} + \bar{\mathbf{z}}_{i-1} \ddot{q}_i + \dot{\boldsymbol{\omega}}_{i-1} \times (\bar{\mathbf{z}}_{i-1} \dot{q}_i), & \text{si } i \text{ es rotacional} \\ \dot{\boldsymbol{\omega}}_{i-1}, & \text{si } i \text{ es prismático} \end{cases}$$

$$\dot{\mathbf{v}}_i = \begin{cases} \dot{\boldsymbol{\omega}}_i \times \mathbf{p}_i^* + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{p}_i^*) + \dot{\mathbf{v}}_{i-1}, & \text{si } i \text{ es rotacional} \\ \bar{\mathbf{z}}_{i-1} \ddot{q}_i + \dot{\boldsymbol{\omega}}_i \times \mathbf{p}_i^* + 2\boldsymbol{\omega}_i \times (\bar{\mathbf{z}}_{i-1} \dot{q}_i) + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \mathbf{p}_i^*) + \dot{\mathbf{v}}_{i-1}, & \text{si } i \text{ es prismático} \end{cases}$$

$$\bar{\mathbf{a}}_{ci} = \dot{\boldsymbol{\omega}}_i \times \bar{\mathbf{s}}_i + \boldsymbol{\omega}_i \times (\boldsymbol{\omega}_i \times \bar{\mathbf{s}}_i) + \dot{\mathbf{v}}_i$$

Ecuaciones hacia atrás: $i = n, n - 1, \dots, 1$

$$\mathbf{F}_i = m_i \bar{\mathbf{a}}_{ci}$$

$$\mathbf{N}_i = \mathbf{I}_i \dot{\boldsymbol{\omega}}_i + \boldsymbol{\omega}_i \times (\mathbf{I}_i \boldsymbol{\omega}_i)$$

$$\mathbf{f}_i = \mathbf{F}_i + \mathbf{f}_{i+1}$$

$$\mathbf{n}_i = \mathbf{n}_{i+1} + \mathbf{p}_i^* \times \mathbf{f}_{i+1} + (\mathbf{p}_i^* + \bar{\mathbf{s}}_i) \times \mathbf{F}_i + \mathbf{N}_i$$

Por lo tanto, las fuerzas o torques aplicadas, sin considerar serán:

$$\tau_i = \begin{cases} \mathbf{n}_i^T \bar{\mathbf{z}}_{i-1} + b_i \dot{q}_i, & \text{si } i \text{ es rotacional} \\ \mathbf{f}_i^T \bar{\mathbf{z}}_{i-1} + b_i \dot{q}_i, & \text{si } i \text{ es prismático} \end{cases}$$

El efecto de la gravedad se puede agregar a los torques sobre los eslabones tomando como aceleración lineal la magnitud del vector de gravedad pero en dirección opuesta. Esta aceleración es ficticia y se añade con la finalidad de producir los mismos efectos en los eslabones que los que produce la gravedad. Por tanto las condiciones iniciales del sistema en nuestro robot son $\bar{\mathbf{a}}_{ci} = -[0, 0, -9.81]^T$

Estas ecuaciones se han implementado en la función de Matlab “MDI_NE.m” que se muestra en el anexo 9.2.6.2.

4.6.4 Dinámica del robot usando Toolbox Robótica

Actualmente existen librerías en Matlab como un método para facilitar todos los cálculos presentes en la descripción de manipuladores. Una de ellas es el ‘*Toolbox de Robótica by Peter Corke*’. Es decir, las funciones vienen ya implementadas para realizar, por ejemplo: rotaciones y traslaciones entre sistemas, matrices de transformación homogénea (usando parámetros D-H o D-H modificado), cálculo de la cinemática directa e inversa y modelos dinámicos inversos.

Además permite la creación del robot de manera representativa para visualizar su comportamiento. En este apartado hemos utilizado las diferentes funciones que tiene la librería para la creación del robot y su uso para encontrar los valores numéricos de la fuerza y torques ejercidos en cada articulación.

El primer paso es crear los eslabones y definir sus parámetros inerciales. Para esto hacemos uso de *Link*. A continuación se muestra, a manera de ejemplo como se definió el eslabón 3.

```
L(3)=Link([ 0 0 A3 0 0 -pi/2], 'standard'); % Crea eslabón 3 (convención DH)
L(3).m = 5.7298; % masa
L(3).r = [-0.3095 0 0]; % centro de gravedad
L(3).I = [ 0.0097 0.1549 0.1520]; % Tensor de inercia (Ixx, Iyy, Izz)
L(3).Jm = 0; % Inercia del motor actuador
L(3).G = 1; % Coeficiente de reducción del actuador
L(3).B = 0; % Rozamiento viscoso
L(3).Tc = [0 0]; % Rozamiento de Coulomb
```

Con todos los eslabones definidos, creamos el robot usando *SerialLink*:

```
IMEbot=SerialLink(L, 'name', 'IMEbot');
```

También podemos generar visualmente el robot con el método *plot*:

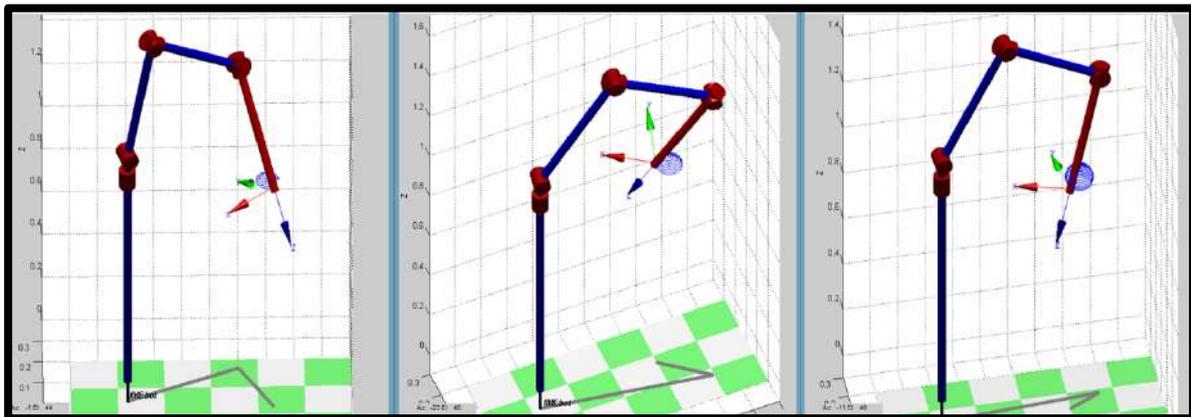


Figura 4.33 IMEbot usando el Toolbox de Robótica

Finalmente utilizamos el método *rne* para el cálculo del torque total, *itorque* para el torque debido a las inercias, *coriolis* para el torque debido a las fuerzas de Coriolis y *gravload* para el torque debido a la gravedad. Se ha implementado en Matlab el uso de esta librería en el script “MDI_Tool.m” que se muestra en el anexo 9.2.6.3.

4.6.5 Prueba de los modelos dinámicos inversos

Con la trayectoria definida (posiciones, velocidades y aceleraciones angulares) y las funciones para el cálculo de los modelos dinámicos inversos, se implementó en Simulink:

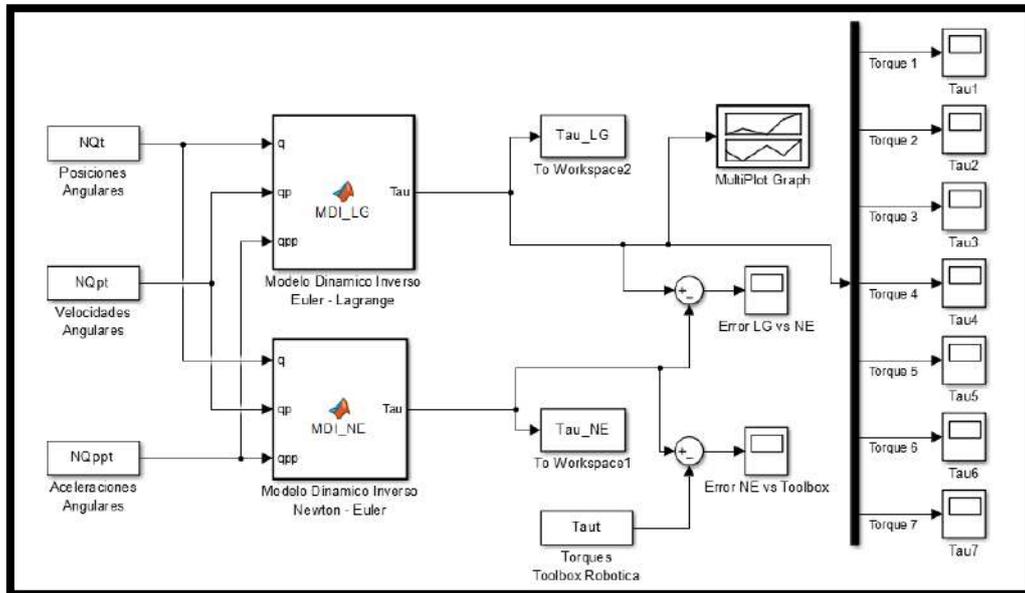


Figura 4.34 Prueba de modelos dinámicos

En el diagrama de la Figura 4.34 se obtienen los torques mediante ambas formulaciones y se comparan entre sí. También se agrega los valores obtenidos con el Toolbox para su comparación. A continuación se muestran los errores obtenidos, los cuales muestran que se llegan exactamente a los mismos resultados.

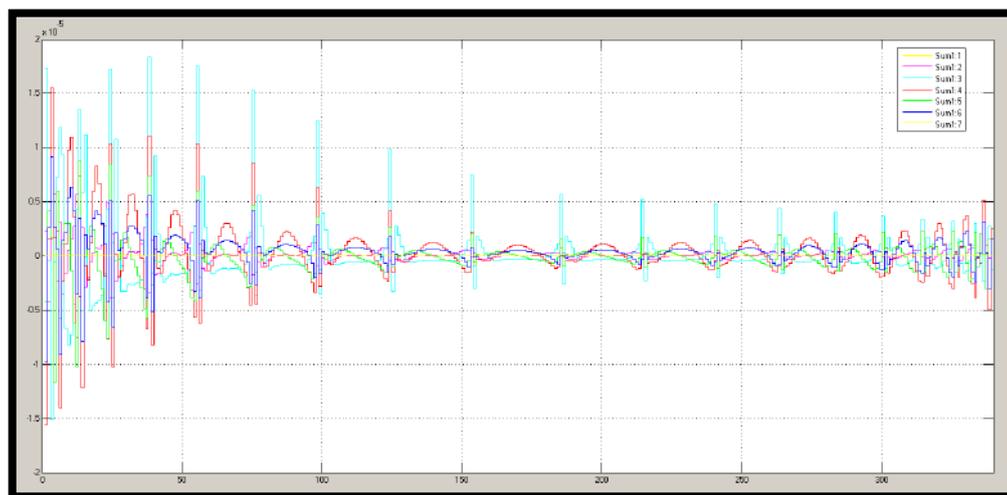


Figura 4.35 Error entre el método Lagrange y Newton-Euler

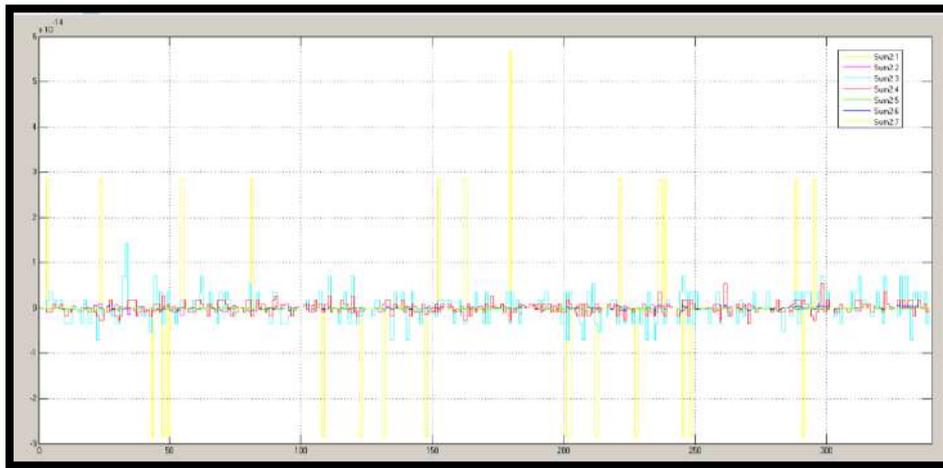


Figura 4.36 Error entre el método Newton-Euler y el proporcionado por el Toolbox de robótica Peter Corke

Por otro lado, en la Figura 4.37 y Figura 4.38 se muestran los valores numéricos de los torques obtenidos. Se puede observar que para la articulación 1 los valores son elevados ya que ésta es la que soporta todo el peso y lidia con el movimiento del robot. En la articulación 2 los valores son pequeños ya que en su movimiento no está afectado por la gravedad y no se están considerando los rozamientos. La articulación 3 y 4 presentan valores medios ya que son los que prácticamente se encargan de posicionar el robot. Los valores de las articulaciones 5, 6 y 7 son acordes a las masas pequeñas de sus eslabones y el movimiento oscilatorio que se requiere.

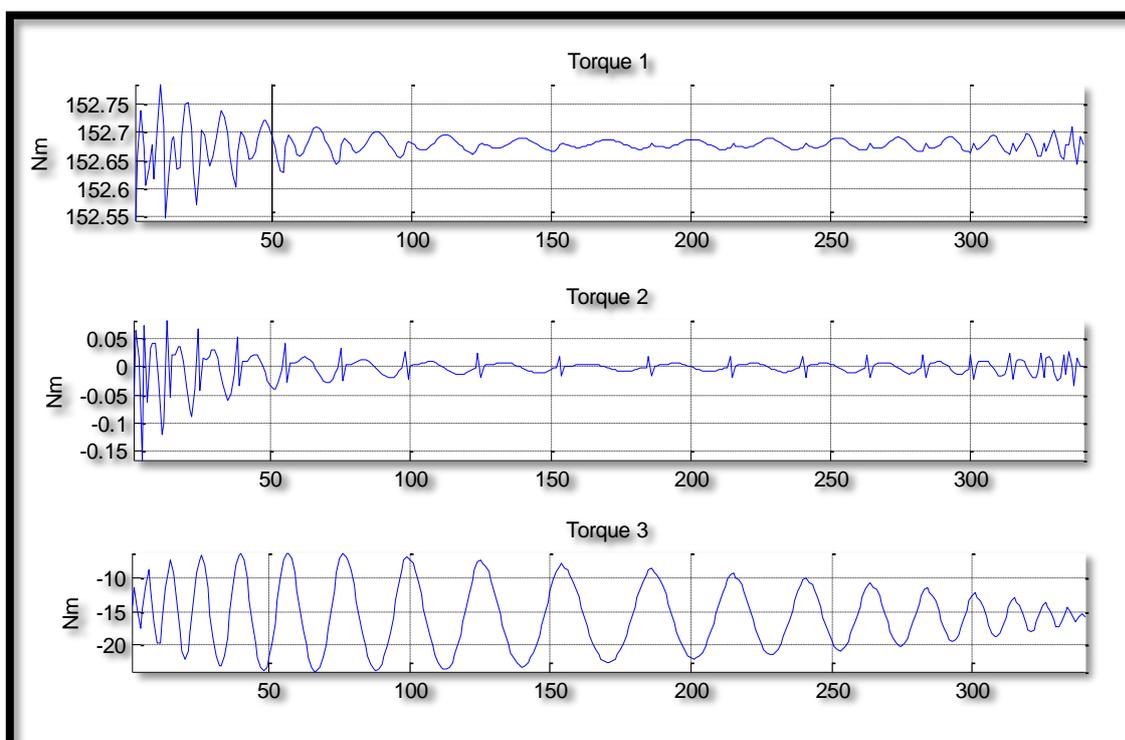


Figura 4.37 Torques en las articulaciones 1, 2 y 3

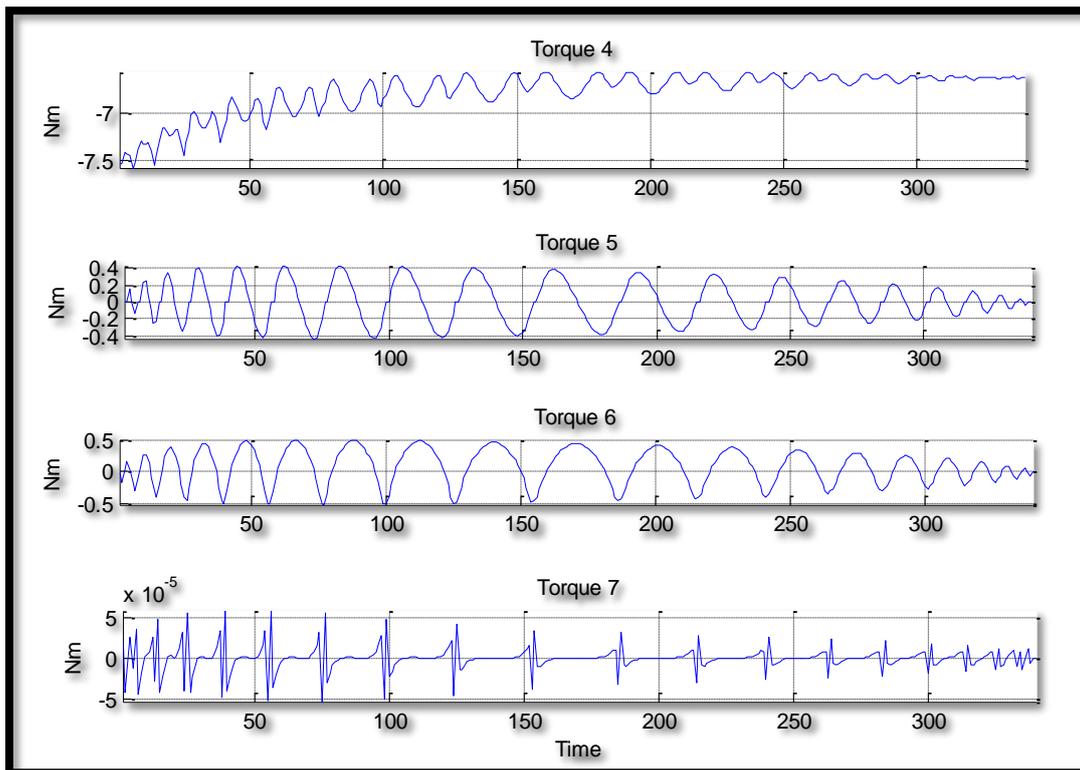


Figura 4.38 Torques en las articulaciones 4, 5, 6 y 7

Capítulo 5

Simulación y Control del Asistente Robótico

Una vez definidas la trayectoria y el tipo de movimiento a realizar por el robot, se debe hacer que el manipular realmente realice estos movimientos. Para esto, emplearemos el *método de control lineal* o llamado también de *articulación independiente*, que es válido cuando el sistema a representar pueda modelarse mediante ecuaciones diferenciales lineales. Sin embargo, a pesar de que los robots son modelados mediante ecuaciones diferenciales no lineales, el control lineal es el que se utiliza con mayor frecuencia en la industria actual. Por esta razón se optó por ello en el presente trabajo.

Se puede apreciar que el control de manipuladores es un problema de control con múltiples entradas y múltiples salidas (MIMO). En este análisis trataremos cada articulación como un sistema independiente que deba controlarse por separado, es decir, de las 7 articulaciones de IMEbot, para 6 de ellas diseñaremos 6 sistemas independientes de control con una entrada y una salida (SISO), ya que la primera articulación es ‘fija’.

Los motores eléctricos de corriente continua son el tipo de actuador más utilizado para los manipuladores. Otro tipo de motores también se han utilizado pero con poca frecuencia en robótica, como son los motores de corriente alterna, debido a la complejidad de su control y los motores de pasos debido a su baja capacidad de torque ^[39].

Se debe tener en cuenta que el diseño mecánico influye en el problema control. Por ejemplo un robot con motores DC de imanes permanentes presenta una dinámica lineal y cuenta con un sistema de engranajes para reducción, los que eliminan en gran medida la inercia acoplada entre las articulaciones, sin embargo introducen fricción, trenes de accionamiento y juego entre ellos.^[38]

En nuestro análisis será necesario obtener el modelo matemático del motor mismo a fin de predecir el comportamiento real del robot y poder ejercer lazos de control que garanticen el seguimiento fiel a la trayectoria propuesta.

5.1 MODELAMIENTO DEL MOTOR DC

En la siguiente figura se muestra el esquema eléctrico y mecánico del motor DC [43]:

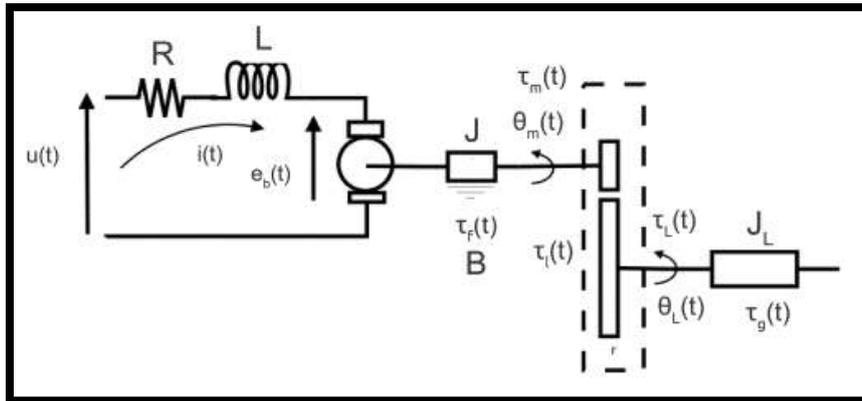


Figura 5.1 Esquema eléctrico y mecánico de un motor dc [43]

Se puede representar mediante las ecuaciones siguientes:

$$u(t) = L \frac{di(t)}{dt} + Ri(t) + e_b(t) \quad (5.1)$$

$$\tau_m(t) = J \frac{d^2\theta_m(t)}{dt^2} + \tau_l(t) + \tau_f(t) \quad (5.2)$$

Donde:

- $u(t)$ la tensión eléctrica aplicada al motor,
- $i(t)$ la corriente eléctrica,
- $\theta_m(t)$ la posición angular del eje del motor,
- $e_b(t)$ la fuerza contra-electromotriz,
- $\tau_m(t)$ el par motor.
- $\tau_l(t)$ el par de la carga visto desde el eje del motor.
- $\tau_f(t)$ el par de fricción.

Los parámetros (L ; R ; J) representan la inductancia, la resistencia eléctrica y el momento de inercia del rotor, respectivamente que consideraremos constantes. Además, el sistema electromecánico satisface las siguientes ecuaciones de acoplo:

$$\tau_m(t) = k_m i(t) \quad (5.3)$$

$$e_b(t) = k_b \dot{\theta}_m(t) \quad (5.4)$$

Donde $k_m [\frac{N}{A}]$ es la constante de par y $k_b [\frac{V\text{-seg}}{rad}]$ la constante de fuerza contra-electromotriz.

Ahora, considerando el modelo clásico de fricción o modelo CVS:

$$\tau_f(t) = \tau_{fc}(t) + \tau_{fv}(t) + \tau_{fs}(t) \quad (5.5)$$

Donde:

- $\tau_{fc}(t)$ el par de fricción seca o de Coulomb.

$\tau_{fV}(t)$ el par de fricción viscosa.

$\tau_{fS}(t)$ el par de fricción estática.

Sabiendo que:

$$\tau_{fC}(t) = \tau_C \operatorname{sgn}(\dot{\theta}_m(t))$$

$$\tau_{fV}(t) = B\dot{\theta}_m(t)$$

$$\tau_{fS}(t) = \begin{cases} \tau_e(t), & |\tau_e(t)| \leq \tau_S, & \dot{\theta}_m(t) = \ddot{\theta}_m(t) = 0 \\ \tau_S \operatorname{sgn}(\tau_e(t)), & |\tau_e(t)| > \tau_S, & \dot{\theta}_m(t) = 0, \ddot{\theta}_m(t) \neq 0 \end{cases}$$

Donde $\tau_e(t)$ representa el par externo:

$$\tau_e(t) = \tau_m(t) - \tau_l(t) + J\ddot{\theta}_m(t)$$

Los parámetros (B ; τ_C ; τ_S) representan las constantes de fricción viscosa, de Coulomb y estática respectivamente. Llamaremos $\tau_C(t)$ a:

$$\tau_C(t) = \tau_{fS}(t) + \tau_{fC}(t) + \tau_l(t) \quad (5.6)$$

Por lo tanto, las ecuaciones serán:

$$u(t) = L \frac{di(t)}{dt} + Ri(t) + k_b \frac{d\theta_m(t)}{dt} \quad (5.7)$$

$$k_m i(t) = J \frac{d^2\theta_m(t)}{dt^2} + B \frac{d\theta_m(t)}{dt} + \tau_C(t) \quad (5.8)$$

Aplicando la transformada de Laplace a la ecuación eléctrica (5.7):

$$\mathcal{L}\{u(t)\} = \mathcal{L}\left\{L \frac{di(t)}{dt} + Ri(t) + k_b \frac{d\theta_m(t)}{dt}\right\}$$

Entonces

$$U(s) = LsI(s) + RI(s) + k_b\theta_m(s) = (Ls + R)I(s) + k_b\theta_m(s) \quad (5.9)$$

Aplicando la transformada de Laplace a la ecuación mecánica (5.8):

$$\mathcal{L}\{k_m i(t)\} = \mathcal{L}\left\{J \frac{d^2\theta_m(t)}{dt^2} + B \frac{d\theta_m(t)}{dt} + \tau_C(t)\right\}$$

Entonces

$$k_m I(s) = Js^2\theta_m(s) + Bs\theta_m(s) + \tau_C(s) = s(Js + B)\theta_m(s) + T_C(s) \quad (5.10)$$

Despejando $I(s)$ de (5.9) y reemplazando en (5.10) se obtiene:

$$\theta_m(s) = \frac{k_m}{s((Ls + R)(Js + B) + k_b k_m)} U(s) - \frac{Ls + R}{s((Ls + R)(Js + B) + k_b k_m)} T_C(s) \quad (5.11)$$

Tal y como se ha considerado en estudios anteriores [44], [45] y [46], podemos asumir que la constante de fricción B es muy pequeña, por lo que la ecuación (5.11) quedaría expresada:

$$\Theta_m(s) = \frac{1}{s} \left(\frac{k_m}{(LJs^2 + RJs + k_b k_m)} U(s) - \frac{Ls + R}{(LJs^2 + RJs + k_b k_m)} T_C(s) \right) \quad (5.12)$$

Si consideramos el principio de superposición, es decir:

$$\theta_m(t) = \theta_u(t) + \theta_{\tau_c}(t) \quad (5.13)$$

Donde:

$\theta_m(t)$ Desplazamiento angular total del eje del motor.

$\theta_u(t)$ Desplazamiento angular del motor sin carga.

$\theta_{\tau_c}(t)$ Desplazamiento angular del motor con carga.

Aplicando la transformada de Laplace a las ecuaciones anterior, tenemos:

$$\Theta_m(s) = \Theta_u(s) + \Theta_{\tau_c}(s)$$

Igualando términos, podemos decir que:

$$G_u(s) = \frac{\Theta_u(s)}{U(s)} = \frac{1}{s} * \frac{k_m}{(LJs^2 + RJs + k_b k_m)} U(s) \quad (5.14)$$

$$G_{\tau_c}(s) = \frac{\Theta_{\tau_c}(s)}{T_C(s)} = -\frac{1}{s} * \frac{Ls + R}{(LJs^2 + RJs + k_b k_m)} T_C(s) \quad (5.15)$$

Donde

$G_u(s)$ Función de transferencia del motor sin carga.

$G_{\tau_c}(s)$ Función de transferencia del motor con carga.

La constante de tiempo eléctrica t_e (en segundos) describe el tiempo de reacción de la corriente al conectar o desconectar un voltaje y está definida por:

$$t_e = \frac{L}{R} \quad (5.16)$$

La constante de tiempo mecánico t_m (en segundos) está definida por:

$$t_m = \frac{RJ}{k_b k_m} \quad (5.17)$$

Utilizando las constantes anteriores, le damos forma a las ecuaciones (5.16) y (5.17):

$$G_u(s) = \frac{\Theta_u(s)}{U(s)} = \frac{1}{s} * \frac{\frac{1}{k_b}}{(t_m t_e s^2 + t_e s + 1)} U(s) \quad (5.18)$$

$$G_{\tau_c}(s) = \frac{\Theta_{\tau_c}(s)}{T_C(s)} = -\frac{1}{s} * \frac{\frac{Ls + R}{k_b k_m}}{(t_m t_e s^2 + t_e s + 1)} T_C(s) \quad (5.19)$$

5.2 BRUSHLESS DC MOTOR

Como en nuestro caso, la intención es la de construir un asistente médico para cirugía laparoscópica, en la que el espacio de trabajo es limitado (para la no obstrucción al médico principal), el mantenimiento no se tiene previsto y se requiere alta precisión, es mejor opción utilizar un motor dc sin escobillas, como veremos a continuación.

El motor de corriente continua sin escobillas (en inglés Brushless, también llamados motores BLDC) a diferencia del motor dc normal, no utiliza escobillas para la conmutación sino que ésta se realiza por control electrónico. Los devanados del estator se energizan en secuencia para crear el campo magnético, además no hay contacto físico entre el rotor y el estator.

| Características | Motor DC Brushless | Motor DC convencional |
|-----------------------|--|--|
| Conmutación | Conmutación electrónica basada en sensores de posición de efecto Hall | Conmutación por escobillas |
| Mantenimiento | Mínimo | Periódico |
| Curva velocidad / par | Plana. Operación a todas las velocidades con la carga definida | Media. A altas velocidades la fricción de las escobillas se incrementa, reduciendo el par |
| Eficiencia | Alta. Sin la caída de tensión por las escobillas | Moderada |
| Torque / tamaño | Alta. Menor tamaño debido a mejores características térmicas porque los bobinados están en el estator por lo que tiene una mejor disipación de calor | Baja. El calor producido en la armadura es disipado en el interior aumentando la temperatura y limitando las características |
| Inercia del motor | Baja debido a los imanes permanentes en el rotor | Alta. Limita las características dinámicas |
| Rango de velocidades | Alta, sin limitaciones mecánicas impuestas por escobillas | Bajo. El límite lo imponen las escobillas |
| Ruido eléctrico | Bajo | Arcos en las escobillas |
| Precio | Alto | Bajo |
| Tipo de control | Complejo | Simple |
| Requisitos de control | Requiere siempre de un control para mantener el motor funcionando. Éste puede usarse también para variar la velocidad. | Sólo se requiere control para la variación de velocidad |

Tabla 5.1 Motor BLDC vs motor DC convencional ^{[45],[47]}

Al no tener un conmutador mecánico o escobillas que se puedan desgastar, eliminan aspectos como el mantenimiento y no producen chispas. Además presentan una menor inercia y fricción en su eje, menor ruido y un mayor torque respecto a su peso, por lo que por lo general son más pequeños que un motor dc con escobillas.

Por otro lado, como el calor se genera en el estator y no en el rotor presentan una mejor disipación del calor. Además, debido a su control electrónico, la regulación de velocidad es precisa y no sufre variaciones con la carga aplicada. Cabe mencionar que tienen un mayor rango de velocidad. Se muestra la tabla comparativa 5.1:

El principio de funcionamiento es a través de sensores de posición del rotor (generalmente sensores hall), el cual contiene el imán permanente, de tal manera que se va energizando convenientemente las bobinas del estator, que está hecho de acero laminado y con ranuras.

El modelo matemático de un motor BLDC no es completamente diferente al de un motor dc convencional. La principal diferencia son los bobinados o fases del estator que pueden configurarse en estrella (Y) o triángulo (Δ), siendo el más común es el de 3-fases.

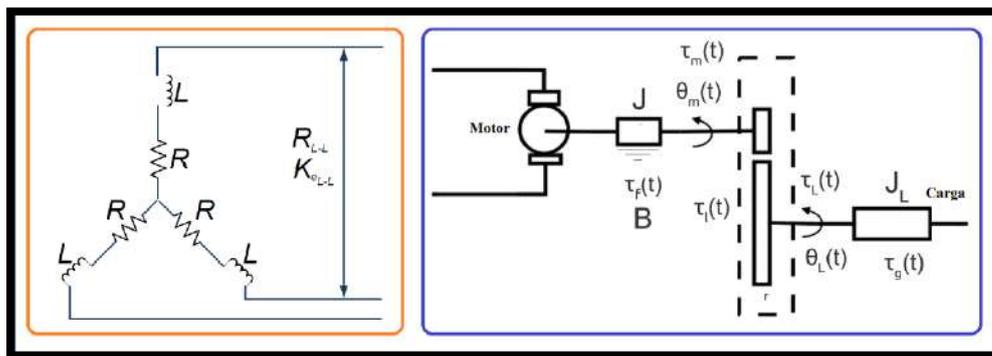


Figura 5.2 Diagrama esquemático de un Brushless DC Motor

La función de transferencia es la misma que la expresada en (5.18) y (5.19), pero los arreglos de las inductancias y resistencias, y de su constante mecánica y eléctrica difieren significativamente debido a la introducción de fases y el tipo de devanado. La constante mecánica y eléctrica para un motor BLDC en conexión estrella está dado por:

$$t_m = \frac{J \sum R_{fase}}{k_b(fase)k_m} \quad (5.20)$$

$$t_e = \frac{L_{l-l}}{\sum R_{fase}} \quad (5.21)$$

Para un arreglo simétrico y con tres fases: $k_b(fase) = \frac{k_b(l-l)}{\sqrt{3}}$, $\sum R_{fase} = 3 R_{fase}$. Por lo tanto las constantes quedan expresadas como:

$$t_m = \frac{J * 3R}{k_b k_m} \quad (5.22)$$

$$t_e = \frac{L}{3R} \quad (5.23)$$

Y las funciones de transferencia serían,

$$G_u(s) = \frac{\Theta_u(s)}{U(s)} = \frac{1}{s} * \frac{\frac{1}{k_b}}{(t_m t_e s^2 + t_e s + 1)} U(s) \quad (5.24)$$

$$G_{\tau_c}(s) = \frac{\Theta_{\tau_c}(s)}{T_c(s)} = -\frac{1}{s} * \frac{\frac{Ls + 3R}{k_b k_m}}{(t_m t_e s^2 + t_e s + 1)} T_c(s) \quad (5.25)$$

5.3 SELECCIÓN DE MOTORES

La selección de los actuadores es un punto muy importante en el diseño de nuestro asistente robótico. Para esto, haremos uso de los valores de los pares calculados en el capítulo anterior a través de la dinámica inversa y utilizaremos las características físicas de los motores que proveen los fabricantes.

Para una selección adecuada del motor es necesario conocer el torque RMS y la velocidad máxima a la que será sometido. Utilizaremos los pares calculados anteriormente y las velocidades de operación. A estos valores le asignaremos un factor de seguridad de 1.5 ^[48].

Los torques RMS se obtuvieron con la función de Matlab $rms(X)$. Donde X es el vector columna de torques para una determinada articulación. Las velocidades se calculan con la función $max(X)$ y $min(X)$ de Matlab. Los valores resultantes fueron:

| Articulación | T_{RMS} (Nm) | T_{RMS} con factor de seguridad 1.5 | Velocidad (RPM) | |
|--------------|----------------|---------------------------------------|-----------------|-----------|
| | | | V_{max} | V_{min} |
| 2 | 0.0226 | 0.0339 | 3.554 | -3.554 |
| 3 | 16.4220 | 24.633 | 6.230 | -0.285 |
| 4 | 6.7936 | 10.1904 | 6.108 | -3.203 |
| 5 | 0.2432 | 0.3648 | 6.280 | -6.280 |
| 6 | 0.2944 | 0.4416 | 1,858 | -11,1 |
| 7 | 0 | 0 | 0 | 0 |

Tabla 5.2 Torques RMS y velocidades máximas

En la Tabla 5.2 no se muestra los valores de la articulación 1 ya que se trata de una articulación regulable manualmente y no tendrá como actuador un motor, sino un diseño mecánico.

Ahora bien, tomando como referencia al robot Da Vinci el cual utiliza motores Maxon ^[49], se ha decidido simular y modelar el comportamiento real de estos motores con sus parámetros técnicos reales para cada articulación. Maxon Motor ofrece un *programa de selección online* con los cuales se puede elegir el motor con las características requeridas.

Los motores seleccionados fueron:

| Artic. | Motor | Reductor | Sensor |
|---------------|---|--|--------------------------------|
| 2 | EC-max 22 Ø22mm, brushless, 25Watt con sensores hall. Part Number: 283858 | Koaxdrive KD 32 Ruido reducido, 17:1 | Encoder MR, tipo M, 512 CPT |
| 3 | EC 45 Ø45mm, brushless, 150Watt con sensores hall. Part Number: 136198 | Planetario GP52, 285:1 | Encoder HEDL 9140 |
| 4 | EC 40 Ø40mm, brushless, 70Watt con sensores hall. Part Number: 283867 | Planetario GP42, 230:1 | Encoder HEDL 5540 |
| 5 | EC-max 22 Ø22mm, brushless, 25Watt con sensores hall. Part Number: 283858 | Koaxdrive KD 32 Ruido reducido, 17:1 | Encoder MR, tipo M, 512 CPT |
| 6 | EC-max 22 Ø22mm, brushless, 25Watt con sensores hall. Part Number: 283858 | Koaxdrive KD 32 Ruido reducido, 33:1 | Encoder MR, tipo M, 512 CPT |
| 7 | RE 13 Ø13mm, Graphite brushes 3Watt | Planetario GP13. 67:1 | Encoder MEnc, 16 CPT |

Tabla 5.3 Motores DC Brushless seleccionados.

En todos los casos se selecciona también los reductores, con la finalidad de aumentar el par entregado por el motor y reducir la velocidad de giro de su eje. Se debe notar que se ha sobredimensionado el motor para la articulación 2, coincidiendo con los seleccionados para las articulaciones 5 y 6, esto debido a que en los pares calculados no se ha tomado en cuenta las fuerzas de fricción. Sin embargo el reductor escogido si difiere, siendo estos de 17:1 y 33:1.

Además, para el motor de la articulación 7 se ha escogido uno de escobillas de grafito, el cual está acoplado mediante un reductor planetario GP13, y que también es utilizado en el robot Da Vinci. Es el único motor de escobillas que se considera, ya que sólo manipula el

giro del laparoscopio. Se podría optar también por uno sin sensor, ya que debería ser accionado directamente por el médico de acuerdo a su requerimiento.

5.4 MOTORES DC MAXON BRUSHLESS

Los motores seleccionados en este trabajo se muestran en la Tabla 5.3. Los parámetros relevantes usados en el modelamiento a lazo abierto fueron extraídos de las hojas de datos técnicos de los motores que provee el fabricante. A continuación se muestran estos datos

| Maxon Motor: EC – max 22 Ø22mm | | |
|---------------------------------------|-------------------|--------------|
| Características | Unidad | Valor |
| Resistencia terminal fase a fase | Ω | 3.44 |
| Inductancia terminal fase a fase | mH | 0.182 |
| Constante de torque | m Nm/A | 17.4 |
| Constante de tiempo mecánico | ms | 5.06 |
| Inercia del rotor | g cm ² | 4.45 |
| Número de fases | - | 3 |

Tabla 5.4 Motor Maxon EC-max 22

| Maxon Motor: EC 45 Ø45mm | | |
|----------------------------------|-------------------|--------------|
| Características | Unidad | Valor |
| Resistencia terminal fase a fase | Ω | 0.936 |
| Inductancia terminal fase a fase | mH | 0.275 |
| Constante de torque | m Nm/A | 37.1 |
| Constante de tiempo mecánico | ms | 8.08 |
| Inercia del rotor | g cm ² | 119 |
| Número de fases | - | 3 |

Tabla 5.5 Motor Maxon EC 45

A modo de ejemplo se mostrará el modelo a lazo abierto del motor Maxon EC 40 de la articulación 4 (Figura 5.3). Los datos de la Tabla 5.6 se reemplazan en las ecuaciones (5.24) y (5.25). El motor sin carga debería alcanzar las 8040 rpm, según su hoja técnica.

| Maxon Motor: EC 40 Ø40mm | | |
|----------------------------------|-------------------|--------------|
| Características | Unidad | Valor |
| Resistencia terminal fase a fase | Ω | 1.35 |
| Inductancia terminal fase a fase | mH | 186 |
| Constante de torque | m Nm/A | 28 |
| Constante de tiempo mecánico | ms | 8.82 |
| Inercia del rotor | g cm ² | 51.2 |
| Número de fases | - | 3 |

Tabla 5.6 Motor Maxon EC 40

Los datos ingresados y el diagrama de bloques implementado en Matlab es:

```
% Motor Articulación 4
% -----
% Tipo de motor: EC 40 Ø40 mm, brushless, 70Watt sensores hall.
% Part Number: 283867
% Reductor: Planetario GP42, 230:1
% Sensor: Encoder HEDL 5540
R4=1.35;      % ohms, Resistencia fase a fase
L4=0.186e-3; % henrios, Inductancia fase a fase
Km4=28e-3;   % Nm/A, Constante de torque
tm4=8.82e-3; % seg., Constante de tiempo mecánico
J4=51.2e-7;  % Kgm^2, Inercia del rotor
f=3;         % número de fases
te4=L4/(f*R4); % seg., Constante de tiempo eléctrico
Kb4=(R4*J4)/(tm4*Km4); % Volt, Fuerza contra-electromotriz
```

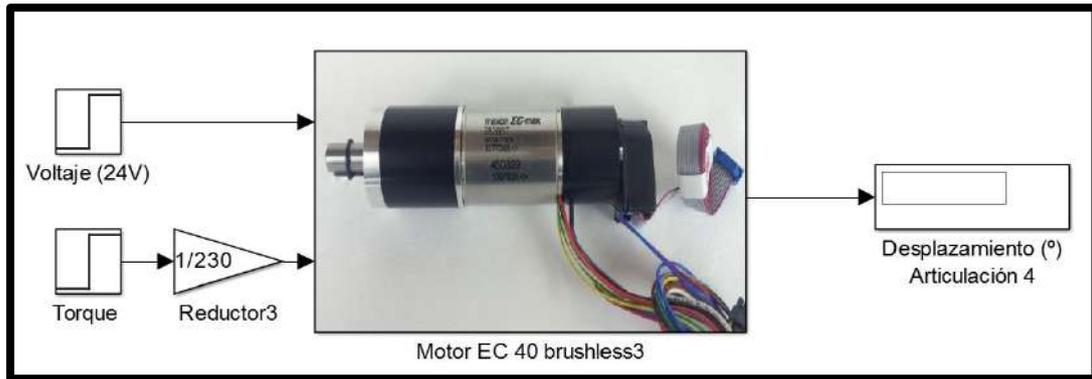


Figura 5.3 Modelo a lazo abierto motor Maxon EC40

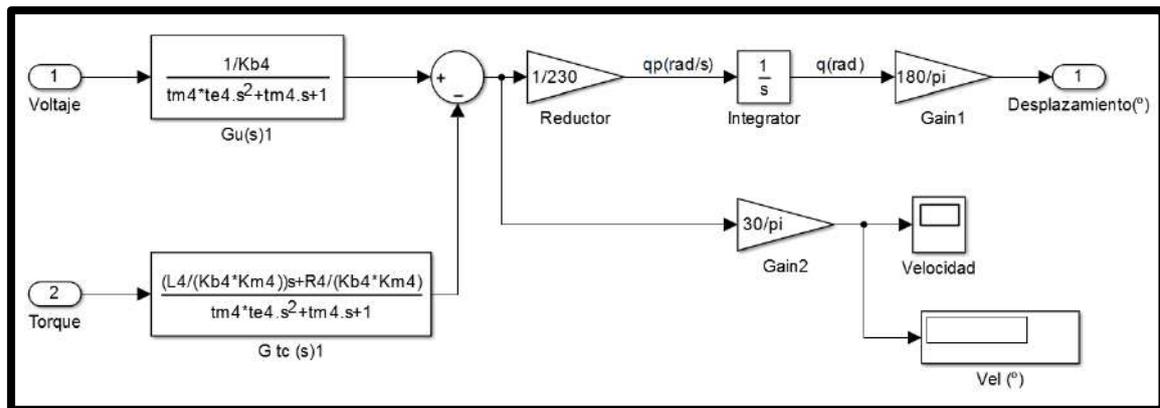


Figura 5.4 Diagrama de bloques Simulink motor Maxon EC40

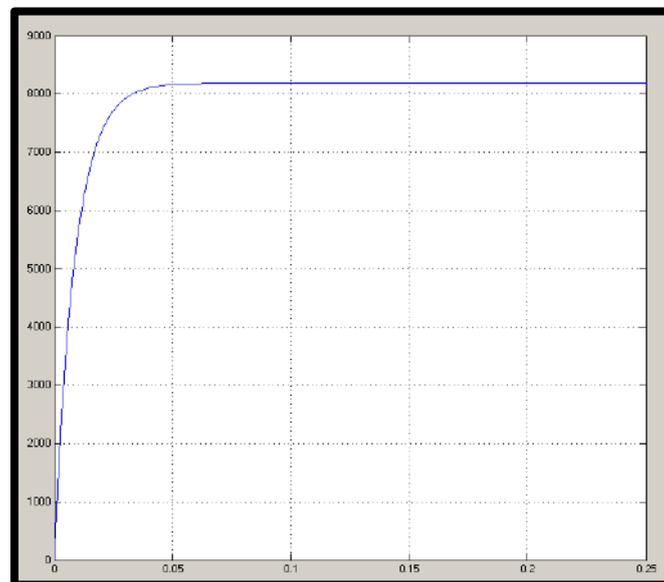


Figura 5.5 Velocidad Motor Maxon EC 40 sin carga

En la Figura 5.5 se puede apreciar que efectivamente el modelo alcanza una velocidad muy cercana a las 8040 rpm que se especifica para el motor sin carga (Torque aplicado = 0 Nm). Este mismo procedimiento se realizó para los motores Maxon EC-max22 y EC45 utilizando los datos de las tablas Tabla 5.4 y Tabla 5.5 respectivamente.

5.5 CONTROL DE MOTORES BRUSHLESS

La estructura de nuestro asistente robótico consta de 5 motores Brushless y 1 con escobillas, para los cuales se ha decidido emplear controladores PID. Se ha diseñado inicialmente el controlador considerando que los efectos de cada grado de libertad no influyen en los otros y se representan como una perturbación.

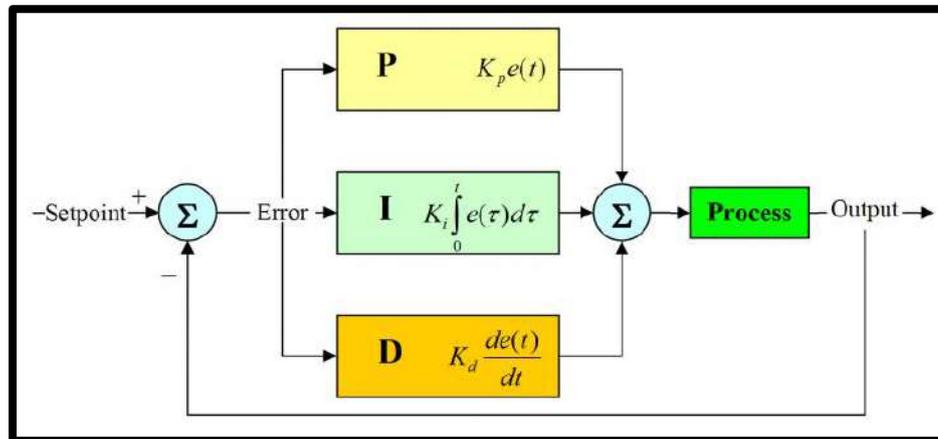


Figura 5.6 Esquema de control PID (Fuente: <http://blascarr.com/arduino-pid-control/>)

La sintonización de los controladores Proporcional - Integral - Derivativo o simplemente controladores PID, consiste en la determinación del ajuste de sus parámetros (K_c , T_i , T_d), para lograr un comportamiento del sistema de control aceptable y robusto de conformidad con algún criterio de desempeño establecido.

El diseño del controlador nos permitirá cumplir básicamente con los siguientes objetivos:

- Seguimiento al valor de la referencia (Set Point)
- Mejorar la dinámica del sistema, es decir, mejorar el tiempo de establecimiento
- Rechazar las perturbaciones

El controlador PID recibe una señal de entrada (generalmente es el error $e(t)$) y proporciona una salida (acción de control, $u(t)$). La señal de control queda determinada por:

$$u(t) = K_p \left(e(t) + \frac{1}{\tau_i} \int_{-\infty}^t e(\tau) d\tau + \tau_d \frac{de(t)}{dt} \right) \quad (5.26)$$

Donde K_p es la ganancia proporcional, τ_i el tiempo integral y τ_d es el tiempo derivativo. La función de transferencia del controlador PID en su forma paralela es:

$$G_{paralelo} = P \left(1 + I \frac{1}{s} + D \frac{N}{1 + N \frac{1}{s}} \right) \quad (5.27)$$

Los parámetros del controlador son representados por la parte proporcional P , la parte integral I , la parte derivativa D y un filtro para ruidos a alta frecuencia en el derivador, donde N depende del tipo de ruido. En nuestro caso se ha utilizado el bloque PID que nos proporciona el entorno Simulink y que nos permite sintonizarlo. Por otro lado, también se debe tener en cuenta las limitaciones físicas en la implementación, ya que en la elección de

los motores, todos ellos tienen como voltaje de entrada 24VDC. Por esta razón se ha colocado un saturador, de manera que la acción de control respete este valor nominal.

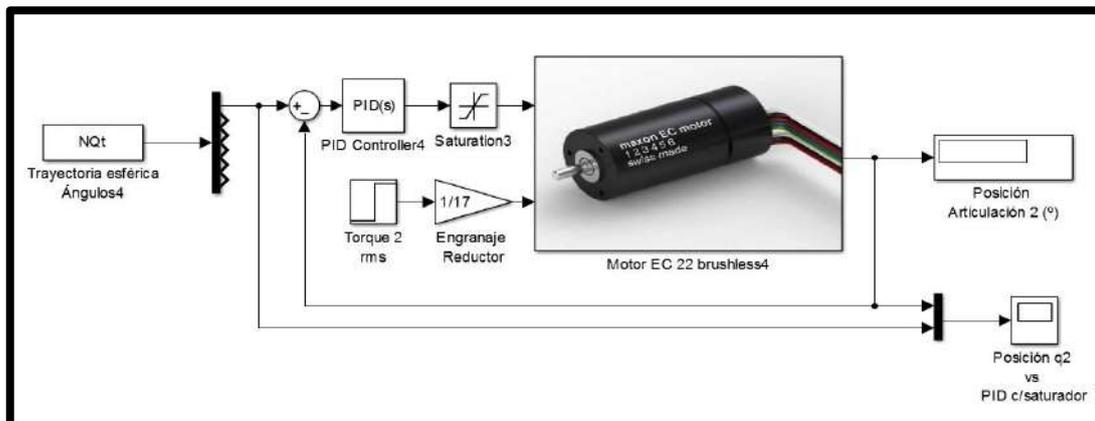


Figura 5.7 Control PID Motor Maxon EC22 con saturador (articulación 2)

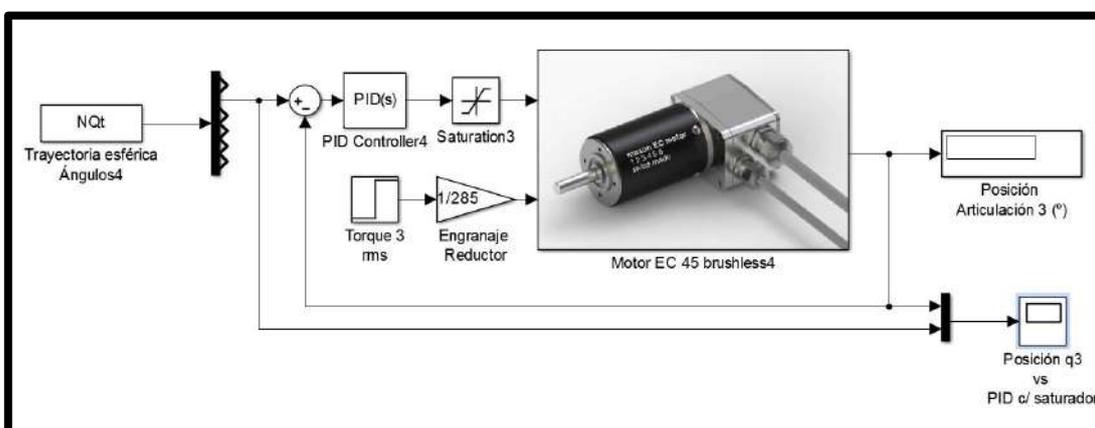


Figura 5.8 Control PID Motor Maxon EC45 con saturador (articulación 3)

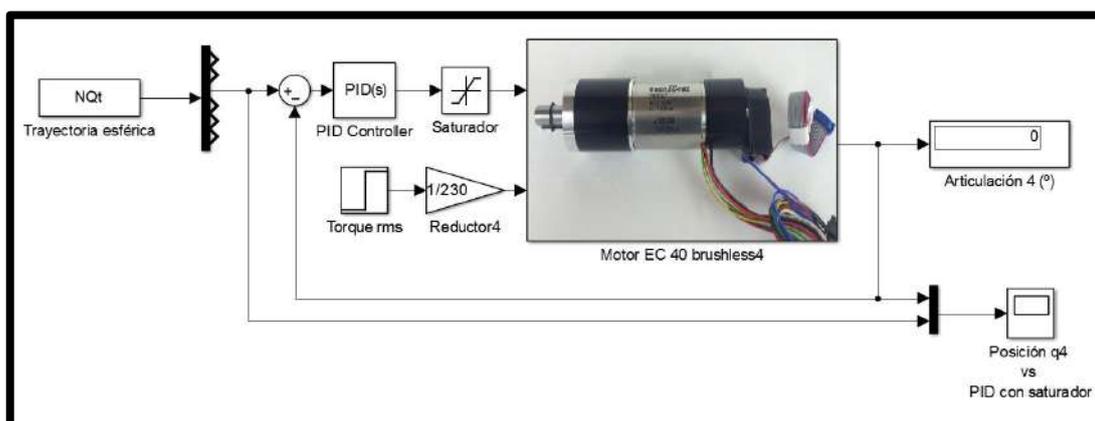


Figura 5.9 Control PID Motor Maxon EC40 con saturador (articulación 4)

Para comparar la respuesta de los controladores incluyendo o no el saturador se muestra la Figura 5.10. En la respuesta al seguimiento de la trayectoria de la articulación 4 (línea azul) se observa que cuando carece de saturador la respuesta es mucho más precisa, sin embargo con saturador el margen de error es despreciable, aparte de garantizar su funcionamiento correcto dentro de su rango de voltaje de entrada.

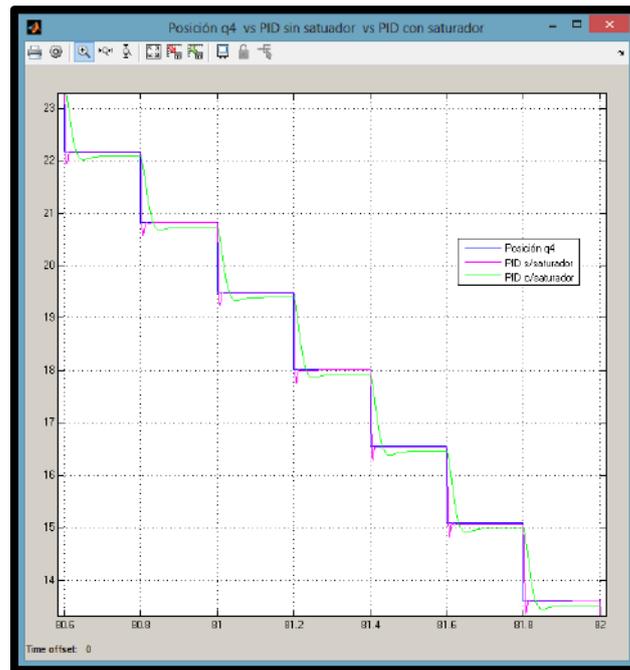


Figura 5.10 Control PID Motor Maxon EC40 con saturador vs sin saturador

De igual manera se hizo la comparación entre el control PID y sus diferentes variantes, es decir, con un control PI y uno PD en la misma articulación. También se probó el sistema incluyendo la carga para el motor y los resultados han sido esencialmente los mismos, ya que debido al reductor incluido el efecto en el eje del motor es muy poco y se considera como una perturbación. Los valores hallados de los parámetros del controlador PID en forma paralela de la ecuación 5.27 para los motores Maxon fueron:

| <i>Parámetros</i> | <i>PID</i> | <i>PI</i> | <i>PD</i> |
|-------------------|------------|-----------|-----------|
| P | 2.01 | 0.60 | 1.85 |
| I | 39.8 | 1.07 | - |
| D | 0.0077 | - | 0.008 |

Tabla 5.7 Parámetros del controlador Motor Maxon EC max 22

| <i>Parámetros</i> | <i>PID</i> | <i>PI</i> | <i>PD</i> |
|-------------------|------------|-----------|-----------|
| P | 7.3 | 7.38 | 32.29 |
| I | 7.5 | 7.55 | - |
| D | 0.01 | - | 0.18 |

Tabla 5.8 Parámetros del controlador Moto Maxon EC40

| <i>Parámetros</i> | <i>PID</i> | <i>PI</i> | <i>PD</i> |
|-------------------|------------|-----------|-----------|
| P | 66.9 | 15.86 | 50.24 |
| I | 168.9 | 20.53 | - |
| D | 0.36 | - | 0.24 |

Tabla 5.9 Parámetros del controlador Motor Maxon EC 45

En las Figura 5.11, Figura 5.12 y Figura 5.13 podemos observar las respuestas obtenidas. Notamos que el control PD es el que mejor en estos casos, ya que tanto en el PID como en el PI debido a su acción integral no sigue fielmente la trayectoria, siendo el control PI el más alejado de nuestros valores de consigna y el que mayor energía (voltaje) requiere.

Se debe mencionar también que la forma de calcular los torques mediante los modelos dinámicos anteriores, también nos permitirían implementar otro tipos de control como: Control PD con compensación de la gravedad y Control por Par Calculado.

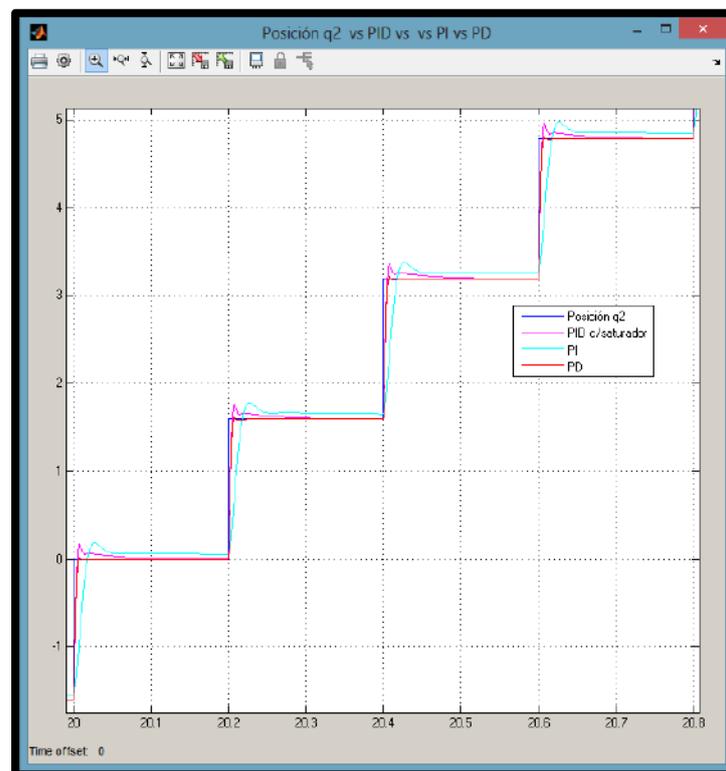


Figura 5.11 Control Motor Maxon EC max 22: PID vs PI vs PD

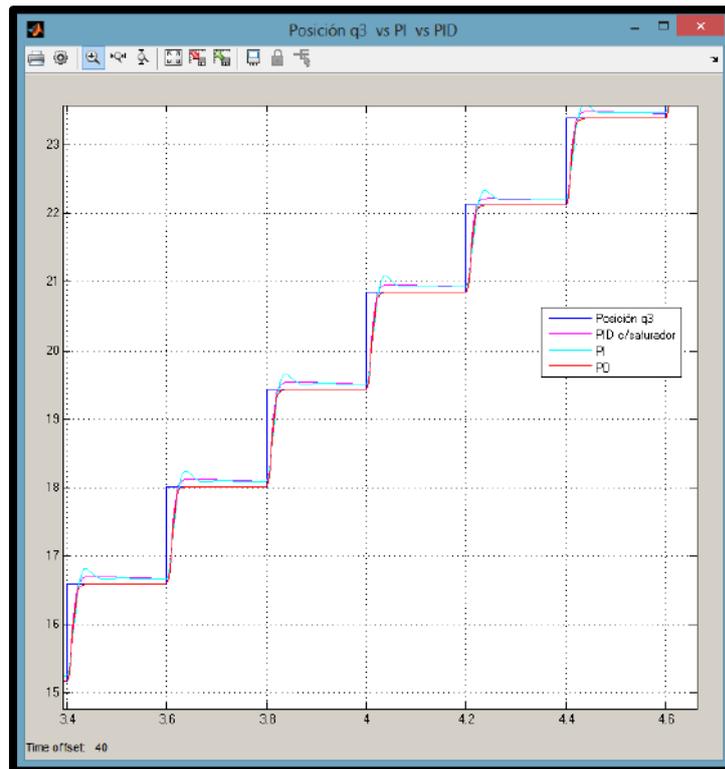


Figura 5.12 Control Motor Maxon EC 45: PID vs PI vs PD

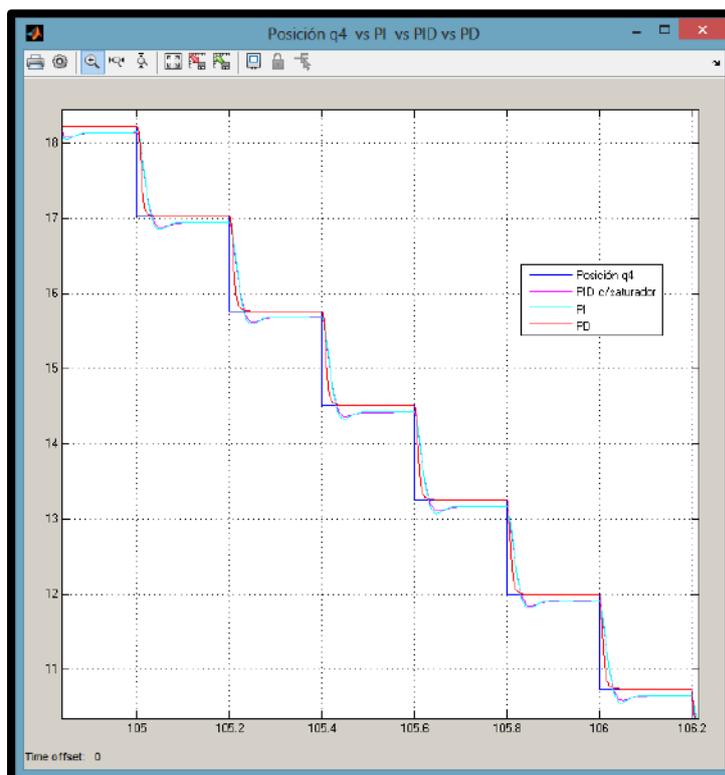


Figura 5.13 Control Motor Maxon EC40: PID vs PI vs PD

5.6 SIMULACIÓN (SIMMECHANICS)

Debido a la necesidad de poder visualizar el comportamiento del asistente robótico y corroborar nuestros modelos matemáticos se empleó la herramienta de simulación SimMechanics (ahora llamado Simscape Multibody) para poder simular los eslabones en movimiento, incluyendo los parámetros físicos reales del robot, tales como: sus dimensiones, materiales y masas.

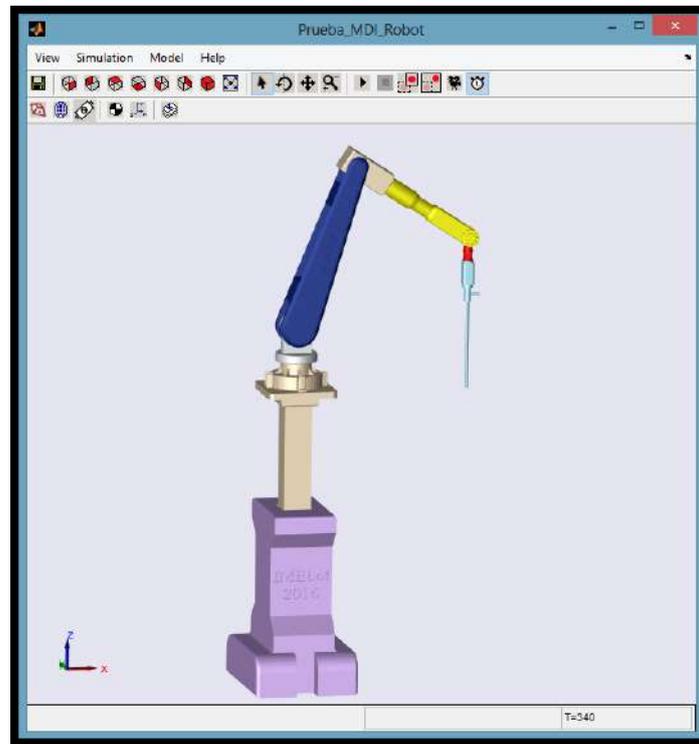


Figura 5.14 Modelo importado de Solidworks

El modelo ha sido importado desde SolidWorks como *first generation* y se muestra en la Figura 5.14. Está compuesto por los siguientes tipos de bloques:

- **Body:**

Son los componentes principales y representan a los cuerpos rígidos, en este caso a los eslabones del robot. En este bloque queda definido la masa, el tensor de inercia, las coordenadas del origen de la pieza y de los sistemas de coordenadas, además de su centro de gravedad.

- **Joints:**

Representan los grados de libertad (ejes) de un eslabón con respecto al otro. Pueden ser rotacionales o prismáticos. Se le pueden añadir puertos para los sensores y actuadores.

- **Constraints:**

Son los bloques que limitan el movimiento mediante ciertas restricciones, por ejemplo el bloque WELD no representa ningún grado de libertad, sino que representa el contacto entre un bloque con el siguiente.

- **Ground:**

Representa un punto fijo que servirá como referencia para el ensamble general del sistema y va conectado al *Machine Environment*.

- **Machine Environment:**

Que es donde se configura las propiedades del entorno de simulación mecánico: el eje en el que es ejercido la gravedad y el modo de análisis (dinámica directa, inversa, cinemática y trimming).

- **Sensors and Actuators:**

Los sensores permiten medir el movimiento de alguna articulación (torques, desplazamiento, velocidad y aceleración angular), mientras que los actuadores proporcionan la señal para producir movimiento (se puede ingresar valores angulares o fuerzas generalizadas).

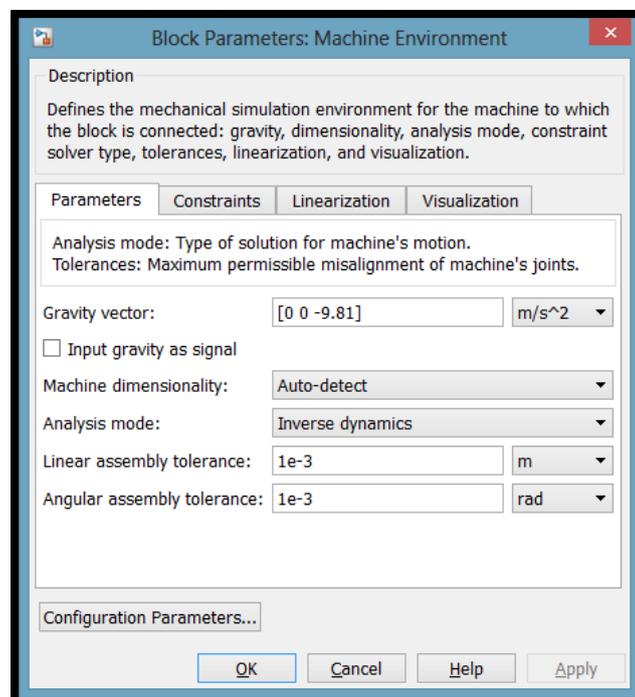


Figura 5.15 Bloque Machine Environment

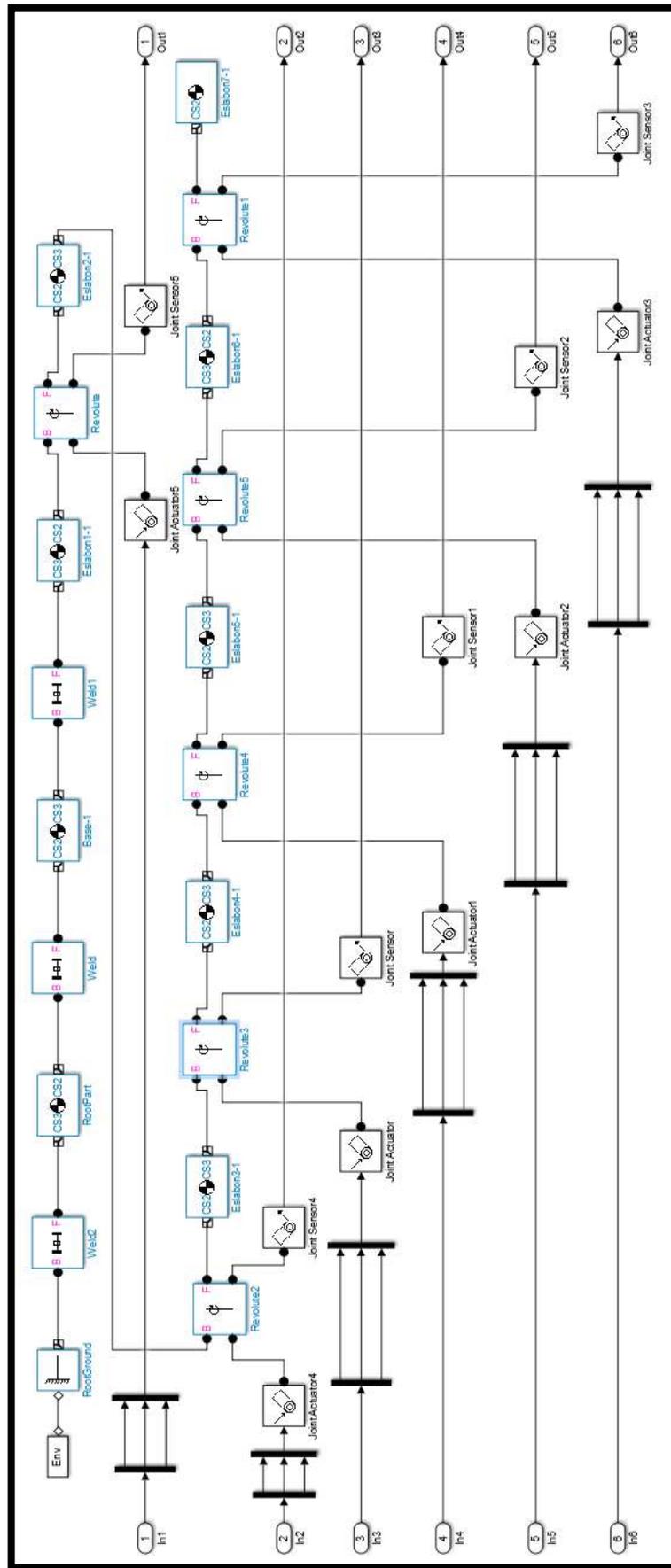


Figura 5.16 Diagrama de bloques del robot IMEbot

Por tanto al diagrama general generado, que se muestra en la Figura 5.16, se le ha añadido los bloques de sensores y actuadores y se le han incorporado los subsistemas de los motores, a fin de simular nuestro robot lo más real posible. Además, se ha fijado los valores de la posición, velocidad y aceleración angular. En la Figura 5.18 se muestran los errores de posición que por ser muy pequeños se consideran bastante aceptables.

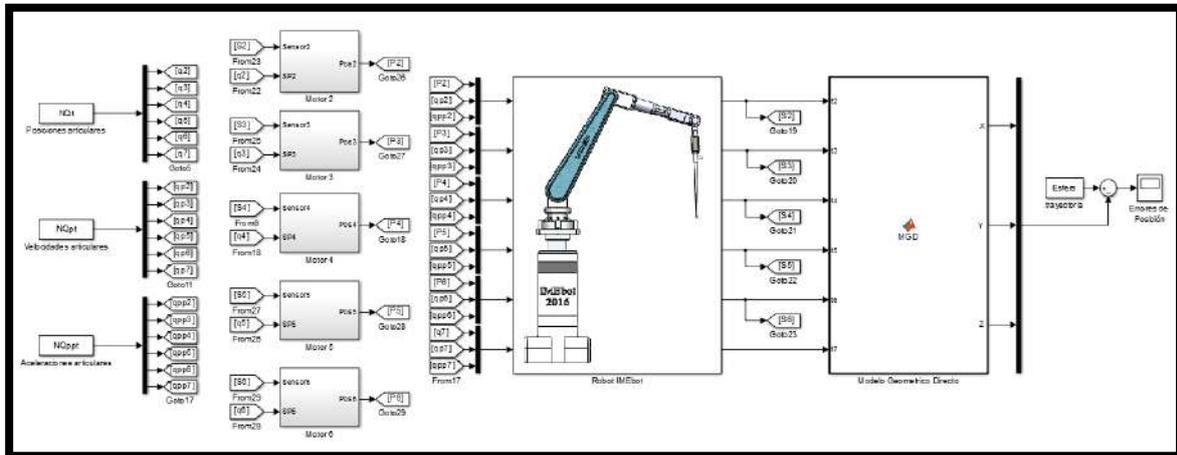


Figura 5.17 Modelo SimMechanics incluyendo control de motores

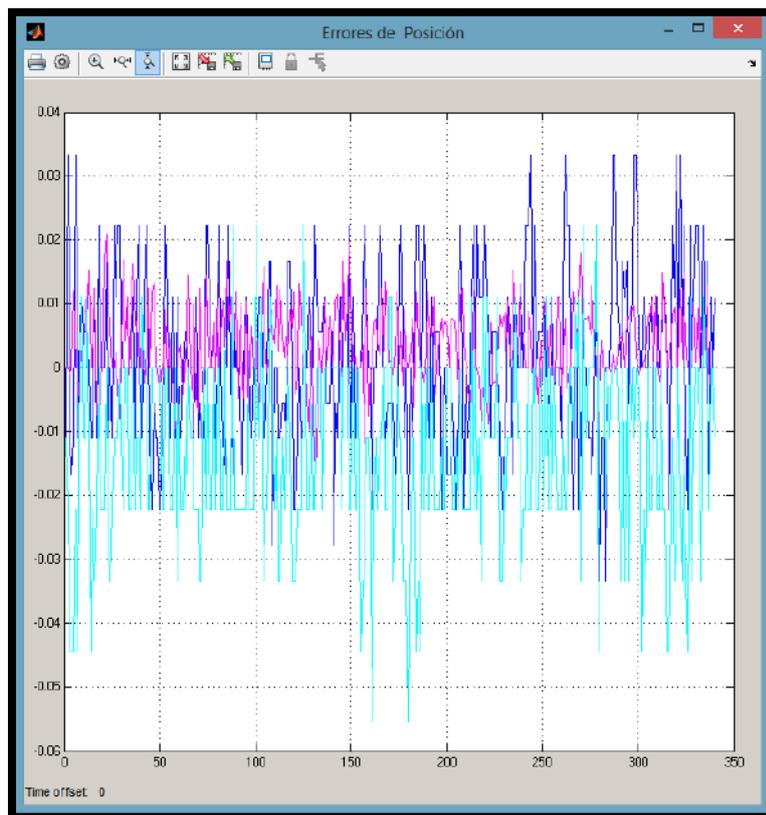


Figura 5.18 Errores de posición

Conclusiones

- Para el modelamiento geométrico directo se puede hacer uso de las distintas convenciones como son los parámetros Denavit-Hartenberg y Khalil-Kleinfinger que nos permiten relacionar la posición entre el espacio de trabajo y un punto de referencia fijo (en nuestro caso, la base).
- En el desarrollo del modelo geométrico inverso se pudo corroborar que el uso de métodos numéricos (método de la bisección) presenta una mayor carga computacional y tiene un tiempo de ejecución mucho mayor en comparación a cuando se encuentra una ecuación cerrada para el cálculo de los ángulos, como lo fue en el desacoplando cinemáticamente el robot.
- El desarrollo de un modelo detallado con el software de diseño SolidWorks nos permite editar los materiales de las piezas y calcular los valores numéricos para los tensores de inercias, las masas, las coordenadas de los centros de gravedad y los momentos principales de inercia para cada eslabón.
- Los modelos dinámicos permiten relacionar las fuerzas y torques ejercidas por un actuador (motor) y el movimiento del eslabón (posiciones, velocidades y aceleraciones angulares). Existen varios métodos para su cálculo como el de Euler – Lagrange y Newton- Euler, además de poder utilizar herramientas de simulación como el complemento SimMechanics.
- El cálculo de los pares ejercidos en cada articulación y las velocidades requeridas en el seguimiento de la trayectoria nos permite calcular sus valores RMS, los cuales sirven para seleccionar los motores comerciales adecuados y sus acoplamientos (engranajes). Éstos son capaces de entregar dichos torques y alcanzar las velocidades requeridas.
- La elección de usar motores Brushless respecto a los motores convencionales DC se basa en aspectos como el tamaño (es menor), no requiere mantenimiento (ya que no presenta escobillas), menor ruido y tienen alta precisión. Además se eligió modelar motores de la marca Maxon ya que son los empleados en el robot Da Vinci.

- El control PD fue el que obtuvo mejor respuesta al seguimiento de la trayectoria esférica propuesta. Se implementó el modelamiento y control para los tres tipos de motores Brushless en Simulink. Además se tuvo en cuenta que el voltaje de alimentación sea el mismo para todos ellos y que el controlador no exija un voltaje mayor a este (se colocó un saturador).
- Con la finalidad de validar nuestros modelamientos, se importó el ensamblaje realizado en SolidWorks al entorno de Matlab, mediante su herramienta SimMechanics. Aquí se incluyó en subsistemas el control de cada motor y el modelo geométrico inverso para comprobar los errores de posición, los cuales, finalmente resultaron despreciables.
- Por último, cabe señalar que el presente trabajo sienta las bases teóricas para la implementación de un asistente robótico prototipo para la manipulación del laparoscopio para cirugía laparoscópica. Tanto los conocimientos para su construcción, así como la voluntad de un centro de entrenamiento de cirugía laparoscópica en nuestra ciudad para el uso del mismo están latentes, así que el proyecto es totalmente viable.

Bibliografía

- [1] KAVOUSSI LR, MOORE RG, ADAMS JB, PARTIN AW. “*Comparison of robotic versus human laparoscopic camera control.*” Journal Urol 154(6):2134–2136. Erratum in: J Urol 1997, 158(4):1530
- [2] OCTAVIO A. CASTILLO; RAFAEL SÁNCHEZ SALAS. “*Bases laparoscópicas de la cirugía robótica.*” Revista Cirugía Robótica en Urología. Arch. Esp. Urol., 60, 4 (357-362), 2007.
- [3] MORENO C.; LOUREIRO C. “*Cirugía Robótica.*”. Monografías de la Asociación Española de Cirujanos N°02. Madrid, 2015
- [4] LONG, J.A.; DESCOTES, J.L.; SKOWRON, O. y cols. “*Use of robotics in laparoscopic urological surgery: state of the art.*” Prog. Urol., 16: 3, 2006.
- [5] S. KOMMU; P. RIMINGTON; C. ANDERSON; A. RANE. “*Initial experience with the EndoAssist camera-holding robot in laparoscopic urological surgery.*” Journal of Robotic Surgery, vol.1, pp.133-137, 2007.
- [6] J. M. SACKIER; Y WANG. “*Robotically assisted laparoscopic surgery: From concept to development.*” Journal Surgical Endoscopy (1994) 8:63-66.
- [7] JAQUES DONNEZ; ROLAND POLET. “*Using a Laparoscope Manipulator in Laparoscopic Gynecological Surgery.*” J Surgical Technology International XVII, 187.
- [8] MÉNDEZ RODRIGUEZ, CRISTIAN; TORRES MUÑOZ, VICTOR. Tesis “*Diseño y simulación en 3D de un robot porta endoscopio para cirugía laparoscópica.*” Pág 18, 2010.
- [9] *Dr. Samadi llega a Lima a compartir innovador sistema de cirugía robótica.* (Abril 2014). Recuperado de: http://www.bnp.gob.pe/portalsnp/index.php?option=com_content&view=article&id=2483:dr-david-samadi-en-lima&catid=347:noticias-abril-2014&Itemid=842

- [10] SOKHANVAR; DARGAHI; NAJARIAN; ARBATANI. *“Tactile Sensing and Displays”*. Wiley 2013. Página 171-182.
- [11] INTUITIVE SURGICAL, *“In Vivo Accessories for Minimally Invasive Robotic Surgery”* US. Patente 8551116B2, Octubre 2013 (actualización).
- [12] INTUITIVE SURGICAL, *“Method and Apparatus for performing minimally invasive surgical procedures”* US. Patente 6905460B2, Enero 2005 (actualización).
- [13] SALINAS, SERGIO ALEXANDER, *“Modelado, simulación en 3D y control de un robot para Cirugía Laparoscópica”*. Tesis de maestría, Popayán 2009.
- [14] OBSERVATORIO DE PROSPECTIVA TECNOLÓGICA INDUSTRIAL Y FEDERACIÓN ESPAÑOLA DE EMPRESAS DE TECNOLOGÍA SANITARIA, *“El Futuro de la Cirugía Mínimamente Invasiva”*, Federación Española de Empresas de Tecnología Sanitaria, 2004.
- [15] CAMARILLO, DAVID; y otros. *“Robotic technology in surgery: past, present, and future.”* The American Journal of Surgery 188. Oct 2004.
- [16] AMERICAN SOCIETY OF COLON AND RECTAL SURGEONS, *“Cirugía Laparoscópica”*. Disponible: <https://www.fascrs.org/cirugia-laparoscopica>. Consultado Junio 2016
- [17] MEDLINE PLUS, *“Cirugía laparoscópica - Serie”*. Disponible online: https://medlineplus.gov/spanish/ency/esp_presentations/100166_1.htm. Consultado Junio 2016.
- [18] PATIÑO, JOSÉ FÉLIX; CERVANTES, JORGE. *“Cirugía Laparoscópica y Toracoscópica.”* McGrawHill Interamericana. 1997. México.
- [19] MORALES, CARLOS. *“Estado Actual de la Cirugía General laparoscópica.”* Laboratorios ITALMEX. Vol. 7. 1994.
- [20] CABELLO BENAVENTE, RAMIRO. *“Bases de Laparoscopia Urológica 6ta Ed.”* Fundación Jiménez Díaz, Feb. 2012. Madrid, España
- [21] KOTCHER FULLER, JOANNA. *“Surgical Technology: Principles and Practice, 6th Ed.”* ELSEVIER 2013, p. 535-568.
- [22] TORRES, RICARDO; SERRA, EDGARDO; MARECOS, MARIA. *“Generalidades de la Cirugía Laparoscópica: Equipamiento e Instrumental”*. Cirugía digestive. F. Galindo. <http://www.sacd.org.ar/>. 2011, I-116, Pág. 1-17
- [23] MISHRA RK. *“Libro de Cirugía Laparoscópica Práctica”*. Jaypee – Highlights Publishers, 2010, p. 10-45.
- [24] BISHOFF, JAY T.; KAVOUSSI, LOUIS R. *“Atlas de Cirugía Urológica Laparoscópica”*. Elsevier, España, 2008, pág. 1-23.

- [25] BEGIN G. F. “*Appendicectomie laparoscopique*”. EMC (Elsevier SAS, Paris), Techniques chirurgicales - Appareil digestif, 40-505, 2006.
- [26] EcuRed Conocimiento con todos y para todos. (s/f). Recuperado 10 de noviembre del 2016 de: https://www.ecured.cu/Mesa_Quirúrgica
- [27] SAHA, SUBIR KUMAR. “*Introducción a la Robótica*”. Mc Graw – Hill, 2008. Pág. 76-112.
- [28] KHALIL, W; DOMBRE, E. “Modeling, Performance Analysis and Control of Robot Manipulators”. Iste 2007.
- [29] CRAIG, JHON J., “*Robótica*”. Pearson Educación, México 2006.
- [30] KHALIL W.; KLEINFINGER J.-F., "A new geometric notation for open and closedloop robots". Proc. IEEE Int. Conf on Robotics and Automation, San Francisco, USA, April 1986, p. 1174-1180.
- [31] BARRIENTOS, ANTONIO et al, “*Fundamentos de Robótica*” Mc Graw Hill - 2da Ed. 2007
- [32] SPONG, MARK et al. “*Robot Modeling and Control*”. John Wiley & Sons, Ltd 2006.
- [33] AMTAI T800 Series Technical Manual. Recuperado el 2 de febrero del 2017 de: [http://accmedicalsolution.com/upload/images/Document/Product/3.Surgery%20Group/1%20Operating%20Table/AMTAI/TM-T800-E02_\(Thailand\).pdf](http://accmedicalsolution.com/upload/images/Document/Product/3.Surgery%20Group/1%20Operating%20Table/AMTAI/TM-T800-E02_(Thailand).pdf)
- [34] REYES CORTÉS, F. “*Matlab aplicado a Robótica y mecatrónica*”. Alfaomega-1era Ed. 2012
- [35] PÉREZ MENENDEZ, F. “*Desarrollo de sistema de control para un manipulador de seis grados de libertad*”. Tesis Fin de Master, Universidad de Oviedo.
- [36] MAZO ESPIAU, F. “*Modelado Dinámico y Simulación del robot industrial Staubli TX90*”. Tesis Fin de Master, Universidad de Oviedo.
- [37] FU, K.S; GONZALEZ, R.C.; LEE, C.S.G. “*Robotics: Control, Sensing, Vision and Intelligence*”. McGraw Hill 1987
- [38] SPONG, M.; HUTCHINSON, S.; VIDYASAGAR, M. “*Robot Dynamics and Control*”. 2da Ed. Enero 2004.
- [39] WILLIAMS, R. L. “*Local Performance Optimization for a Class of Redundant Eight-Degree-of Freedom Manipulators*”. NASA Technical Paper 3417, NASA Technical Paper 3417, NASA Langley Research Center, Hampton, VA, marzo de 1994.
- [40] CHAPMAN, STEPHEN. “*Máquinas Eléctricas*”. McGraw Hill 2012, 5ta Edición.
- [41] BRAUN, JAUN. “*Libro de fórmulas – Maxon Academy*”. Editorial maxon academy, Sachseln, 1era edición, 2013.
- [42] Simscape Multibody, MathWorks. Recuperado el 4 de marzo del 2017 de: <https://www.mathworks.com/products/simmechanics.html>

- [43] MONASTERIO, F.; GUTIERREZ, A. “*Modelo lineal de un motor de corriente continua*”. Abril 2012
- [44] SADO, F. et al. “*Independent Joint Control of a 3-DOF Robotic System using PI Controller*”. 5th International Conference on Computer & Communication Engineering, IEEE 2014.
- [45] HERNANDEZ, J.; ROSERO, W. “*Diseño e implementación de un prototipo para apertura de cráneo (craneótomo)*”. Tesis, Universidad de Nariño, 2014.
- [46] OGUNTOYINBO, O. “*PID Control of Brushless DC Motor and Robot Trajectory Planning and Simulation with Matlab/Simulink*”. University of Applied Sciences, 2009.
- [47] PALMA, S.; TORO, G.; VALDIVIESO, C. “*Control de motores sin escobillas (BLDC) y con sensores usando microcontrolador ARM Cortex3 con 32 bits de LpcXpresso*”. Escuela Superior Politécnica del Litoral, Guayaquil – Ecuador.
- [48] *Prácticas de Robótica utilizando Matlab. Práctica 4: Selección de Servoaccionamientos*. Recuperado el 20 de marzo del 2017 de: <http://nbio.umh.es/files/2012/04/practica4.pdf>
- [49] Robots para operaciones mínimamente invasivas. Recuperado el 30 de marzo del 2017 de: <http://www.maxonmotor.es/maxon/view/application/Robots-para-operaciones-minimamente-invasivas>

Glosario

1. Laparoscopia

El laparoscopia es un instrumento óptico que se utiliza para ver el contenido de la cavidad abdominal durante cirugías mínimamente invasivas. El equipo completo consta de una fuente de luz (la cual se transmite hasta el laparoscopia por medio de fibra óptica), un equipo de vídeo con monitores que tiene la posibilidad de registrar el procedimiento en medios magnéticos (videocintas) o DVD y la pieza manual que es la que se introduce en la cavidad abdominal. El laparoscopia se introduce en la cavidad abdominal a través de una pequeña incisión que se hace en la pared abdominal, el sitio de la incisión dependerá del tipo de cirugía que se realizará y de otros factores como la presencia de cicatrices previas).

El laparoscopia es un instrumento reutilizable que debe ser esterilizado tras cada cirugía. Existen instrumentos afines como el toracoscopia (para cirugía torácica) y el artroscopia (para cirugía de las articulaciones) que siguen exactamente los mismos principios y varían tan sólo, en algunos casos, respecto al tamaño.

2. Colecistectomía

La colecistectomía es la intervención quirúrgica que se realiza para extraer una vesícula biliar enferma: vesícula que está infectada (colecistitis), que está inflamada, o que está bloqueada (obstruida) por estar llena de cálculos biliares.

3. Cistectomía

Cistectomía es el término médico utilizado para referirse a la remoción quirúrgica total o parcial de la vejiga urinaria. La condición más común que justifica una cistectomía es el cáncer de vejiga. En todos los casos, es necesario hacer una derivación de las vías urinarias inferiores, lo que usualmente se hace con un conducto ileal o Bricker, lo que significa conectar la vía urinaria con el íleon.

4. –SCOPIA Sufijo que significa examen, exploración, observación.

5. –TOMÍA Sufijo que significa corte, incisión.

6. Endoscopio

Es un instrumento (producto sanitario) en forma de tubo, que puede ser rígido o flexible, y que contiene una luz y una óptica que permiten la visualización del interior de un órgano hueco o de una cavidad corporal. El procedimiento diagnóstico en el que se utiliza algún tipo de endoscopio se llama endoscopia. El endoscopio es un instrumento que se utiliza para observar dentro de una cavidad, conducto u órgano hueco. Está conformado por una sonda flexible la cual presenta una luz que permite que la cavidad sea observable y una cámara en el extremo de la sonda. Se utiliza introduciendo el instrumento a través de un orificio natural del cuerpo como la boca, el recto o la uretra, o también a través de una incisión. El endoscopio es un instrumento que ha permitido el desarrollo y la especificidad en la medicina, influyendo tanto en la prevención, diagnóstico y pronóstico de muchas enfermedades.

7. HEMOSTASIA

La hemostasia es el conjunto de los fenómenos fisiológicos que concurren a la prevención y detención de las hemorragias. Esta participa en la reparación de la brecha vascular y, de manera general, se encarga del mantenimiento de la integridad de los vasos.

8. PERITONEO

Es la membrana serosa que cubre la mayor parte de los órganos del abdomen y forma pliegues que envuelven las vísceras. Tiene 2 capas: la capa exterior, llamada peritoneo parietal, está adherida a la pared abdominal y la capa interior, peritoneo visceral, envuelve los órganos situados dentro de la cavidad abdominal. El espacio entre ambas capas se denomina cavidad peritoneal; contiene una pequeña cantidad de fluido lubricante (alrededor de 50 ml) que permite a ambas capas deslizarse entre sí.

Anexos

9.1 METODO DE PAUL

El método consiste en la pre-multiplicación de las matrices de transformación homogénea inversas:

9.1.1 Iteración 1: $U_6 = {}^6T_7$

➤ Componentes de la matriz de la izquierda $U_6 = {}^6T_5 {}^5T_4 {}^4T_3 {}^3T_2 {}^2T_1 {}^1T_0 * U_0$

$$U_6(1,1) = c6 * (c5 * (c4 * (s3 * sz + c3 * (c2 * sx + s2 * sy))) + s4 * (c3 * sz - s3 * (c2 * sx + s2 * sy))) + s5 * (c2 * sy - s2 * sx)) + s6 * (c4 * (c3 * sz - s3 * (c2 * sx + s2 * sy)) - s4 * (s3 * sz + c3 * (c2 * sx + s2 * sy)))$$

$$U_6(2,1) = c6 * (c4 * (c3 * sz - s3 * (c2 * sx + s2 * sy)) - s4 * (s3 * sz + c3 * (c2 * sx + s2 * sy))) - s6 * (c5 * (c4 * (s3 * sz + c3 * (c2 * sx + s2 * sy))) + s4 * (c3 * sz - s3 * (c2 * sx + s2 * sy))) + s5 * (c2 * sy - s2 * sx))$$

$$U_6(3,1) = s5 * (c4 * (s3 * sz + c3 * (c2 * sx + s2 * sy)) + s4 * (c3 * sz - s3 * (c2 * sx + s2 * sy))) - c5 * (c2 * sy - s2 * sx)$$

$$U_6(1,2) = c6 * (c5 * (s4 * (c3 * nz - s3 * (c2 * nx + ny * s2)) + c4 * (nz * s3 + c3 * (c2 * nx + ny * s2))) + s5 * (c2 * ny - nx * s2)) - s6 * (s4 * (nz * s3 + c3 * (c2 * nx + ny * s2)) - c4 * (c3 * nz - s3 * (c2 * nx + ny * s2)))$$

$$U_6(2,2) = -c6 * (s4 * (nz * s3 + c3 * (c2 * nx + ny * s2)) - c4 * (c3 * nz - s3 * (c2 * nx + ny * s2))) - s6 * (c5 * (s4 * (c3 * nz - s3 * (c2 * nx + ny * s2)) + c4 * (nz * s3 + c3 * (c2 * nx + ny * s2))) + s5 * (c2 * ny - nx * s2))$$

$$U_6(3,2) = s5 * (s4 * (c3 * nz - s3 * (c2 * nx + ny * s2)) + c4 * (nz * s3 + c3 * (c2 * nx + ny * s2))) - c5 * (c2 * ny - nx * s2)$$

$$U_6(1,3) = c6 * (s5 * (ay * c2 - ax * s2) + c5 * (c4 * (az * s3 + c3 * (ax * c2 + ay * s2)) + s4 * (az * c3 - s3 * (ax * c2 + ay * s2)))) + s6 * (c4 * (az * c3 - s3 * (ax * c2 + ay * s2)) - s4 * (az * s3 + c3 * (ax * c2 + ay * s2)))$$

$$U_6(2,3) = c6 * (c4 * (az * c3 - s3 * (ax * c2 + ay * s2)) - s4 * (az * s3 + c3 * (ax * c2 + ay * s2))) - s6 * (s5 * (ay * c2 - ax * s2) + c5 * (c4 * (az * s3 + c3 * (ax * c2 + ay * s2)) + s4 * (az * c3 - s3 * (ax * c2 + ay * s2))))$$

$$U_6(3,3) = s5 * (c4 * (az * s3 + c3 * (ax * c2 + ay * s2)) + s4 * (az * c3 - s3 * (ax * c2 + ay * s2))) - c5 * (ay * c2 - ax * s2)$$

$$U_6(1,4) = c6 * (s5 * (Py * c2 - Px * s2) + c5 * (c4 * (s3 * (Pz - r1) + c3 * (Px * c2 + Py * s2)) - c4 * d3 + s4 * (c3 * (Pz - r1) - s3 * (Px * c2 + Py * s2)))) - s6 * (r5 - d3 * s4 - c4 * (c3 * (Pz - r1) - s3 * (Px * c2 + Py * s2))) + s4 * (s3 * (Pz - r1) + c3 * (Px * c2 + Py * s2)))$$

$$U_6(2,4) = -c6 * (r5 - d3 * s4 - c4 * (c3 * (Pz - r1) - s3 * (Px * c2 + Py * s2)) + s4 * (s3 * (Pz - r1) + c3 * (Px * c2 + Py * s2))) - s6 * (s5 * (Py * c2 - Px * s2) + c5 * (c4 * (s3 * (Pz - r1) + c3 * (Px * c2 + Py * s2)) - c4 * d3 + s4 * (c3 * (Pz - r1) - s3 * (Px * c2 + Py * s2))))$$

$$U_6(3,4) = s5 * (c4 * (s3 * (Pz - r1) + c3 * (Px * c2 + Py * s2)) - c4 * d3 + s4 * (c3 * (Pz - r1) - s3 * (Px * c2 + Py * s2))) - c5 * (Py * c2 - Px * s2)$$

$$U_6(4,1) = U_6(4,2) = U_6(4,3) = 0$$

$$U_6(4,4) = 1$$

➤ Componentes de la matriz de la derecha 6T_7

$${}^6T_7 = \begin{bmatrix} c_7 & -s_7 & 0 & 0 \\ 0 & 0 & -1 & -r_7 \\ s_7 & c_7 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

9.1.2 Iteración 2: $U_5 = {}^5T_7$

➤ Componentes de la matriz de la izquierda $U_5 = {}^5T_4 {}^4T_3 {}^3T_2 {}^2T_1 {}^1T_0 * U_0$

$$U_5(1,1) = c5 * (c4 * (s3 * sz + c3 * (c2 * sx + s2 * sy)) + s4 * (c3 * sz - s3 * (c2 * sx + s2 * sy))) + s5 * (c2 * sy - s2 * sx)$$

$$U_5(2,1) = c_5 * (c_2 * s_y - s_2 * s_x) - s_5 * (c_4 * (s_3 * s_z + c_3 * (c_2 * s_x + s_2 * s_y)) + s_4 * (c_3 * s_z - s_3 * (c_2 * s_x + s_2 * s_y)))$$

$$U_5(3,1) = c_4 * (c_3 * s_z - s_3 * (c_2 * s_x + s_2 * s_y)) - s_4 * (s_3 * s_z + c_3 * (c_2 * s_x + s_2 * s_y))$$

$$U_5(1,2) = c_5 * (s_4 * (c_3 * n_z - s_3 * (c_2 * n_x + n_y * s_2)) + c_4 * (n_z * s_3 + c_3 * (c_2 * n_x + n_y * s_2))) + s_5 * (c_2 * n_y - n_x * s_2)$$

$$U_5(2,2) = c_5 * (c_2 * n_y - n_x * s_2) - s_5 * (s_4 * (c_3 * n_z - s_3 * (c_2 * n_x + n_y * s_2)) + c_4 * (n_z * s_3 + c_3 * (c_2 * n_x + n_y * s_2)))$$

$$U_5(3,2) = c_4 * (c_3 * n_z - s_3 * (c_2 * n_x + n_y * s_2)) - s_4 * (n_z * s_3 + c_3 * (c_2 * n_x + n_y * s_2))$$

$$U_5(1,3) = s_5 * (a_y * c_2 - a_x * s_2) + c_5 * (c_4 * (a_z * s_3 + c_3 * (a_x * c_2 + a_y * s_2)) + s_4 * (a_z * c_3 - s_3 * (a_x * c_2 + a_y * s_2)))$$

$$U_5(2,3) = c_5 * (a_y * c_2 - a_x * s_2) - s_5 * (c_4 * (a_z * s_3 + c_3 * (a_x * c_2 + a_y * s_2)) + s_4 * (a_z * c_3 - s_3 * (a_x * c_2 + a_y * s_2)))$$

$$U_5(3,3) = c_4 * (a_z * c_3 - s_3 * (a_x * c_2 + a_y * s_2)) - s_4 * (a_z * s_3 + c_3 * (a_x * c_2 + a_y * s_2))$$

$$U_5(1,4) = s_5 * (P_y * c_2 - P_x * s_2) + c_5 * (c_4 * (s_3 * (P_z - r_1) + c_3 * (P_x * c_2 + P_y * s_2)) - c_4 * d_3 + s_4 * (c_3 * (P_z - r_1) - s_3 * (P_x * c_2 + P_y * s_2)))$$

$$U_5(2,4) = c_5 * (P_y * c_2 - P_x * s_2) - s_5 * (c_4 * (s_3 * (P_z - r_1) + c_3 * (P_x * c_2 + P_y * s_2)) - c_4 * d_3 + s_4 * (c_3 * (P_z - r_1) - s_3 * (P_x * c_2 + P_y * s_2)))$$

$$U_5(3,4) = d_3 * s_4 - r_5 + c_4 * (c_3 * (P_z - r_1) - s_3 * (P_x * c_2 + P_y * s_2)) - s_4 * (s_3 * (P_z - r_1) + c_3 * (P_x * c_2 + P_y * s_2))$$

$$U_5(4,1) = U_5(4,2) = U_5(4,3) = 0; U_5(4,4) = 1$$

➤ Componentes de la matriz de la derecha 5T_7

$${}^5T_7 = \begin{bmatrix} c_6 * c_7 & -c_6 * s_7 & s_6 & r_7 * s_6 \\ -s_7 & -c_7 & 0 & 0 \\ s_6 * c_7 & -s_6 * s_7 & -c_6 & -c_6 * r_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

9.1.3 Iteración 3: $U_4 = {}^4T_7$

➤ Componentes de la matriz de la izquierda $U_4 = {}^4T_3 {}^3T_2 {}^2T_1 {}^1T_0 * U_0$

$$U_4(1,1) = c_4 * (s_3 * s_z + c_3 * (c_2 * s_x + s_2 * s_y)) + s_4 * (c_3 * s_z - s_3 * (c_2 * s_x + s_2 * s_y))$$

$$U_4(2,1) = c4 * (c3 * sz - s3 * (c2 * sx + s2 * sy)) - s4 * (s3 * sz + c3 * (c2 * sx + s2 * sy))$$

$$U_4(3,1) = s2 * sx - c2 * sy$$

$$U_4(1,2) = s4 * (c3 * nz - s3 * (c2 * nx + ny * s2)) + c4 * (nz * s3 + c3 * (c2 * nx + ny * s2))$$

$$U_4(2,2) = c4 * (c3 * nz - s3 * (c2 * nx + ny * s2)) - s4 * (nz * s3 + c3 * (c2 * nx + ny * s2))$$

$$U_4(3,2) = nx * s2 - c2 * ny$$

$$U_4(1,3) = c4 * (az * s3 + c3 * (ax * c2 + ay * s2)) + s4 * (az * c3 - s3 * (ax * c2 + ay * s2))$$

$$U_4(2,3) = c4 * (az * c3 - s3 * (ax * c2 + ay * s2)) - s4 * (az * s3 + c3 * (ax * c2 + ay * s2))$$

$$U_4(3,3) = ax * s2 - ay * c2$$

$$U_4(1,4) = c4 * (s3 * (Pz - r1) + c3 * (Px * c2 + Py * s2)) - c4 * d3 + s4 * (c3 * (Pz - r1) - s3 * (Px * c2 + Py * s2))$$

$$U_4(2,4) = d3 * s4 + c4 * (c3 * (Pz - r1) - s3 * (Px * c2 + Py * s2)) - s4 * (s3 * (Pz - r1) + c3 * (Px * c2 + Py * s2))$$

$$U_4(3,4) = Px * s2 - Py * c2$$

$$U_4(4,1) = U_4(4,2) = U_4(4,3) = 0$$

$$U_4(4,4) = 1$$

► Componentes de la matriz de la derecha 4T_7

$${}^4T_7 = \begin{bmatrix} s_5 s_7 + c_5 c_6 c_7 & s_5 c_7 - c_5 c_6 s_7 & c_5 s_6 & c_5 s_6 r_7 \\ s_6 c_7 & -s_6 s_7 & -c_6 & r_5 - c_6 r_7 \\ c_5 s_7 - s_5 c_6 c_7 & c_5 c_7 + s_5 c_6 s_7 & -s_5 s_6 & -s_5 s_6 r_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

9.1.4 Iteración 4: $U_3 = {}^3T_7$

► Componentes de la matriz de la izquierda $U_3 = {}^3T_2 {}^2T_1 {}^1T_0 * U_0$

$$U_3(1,1) = s3 * sz + c3 * (c2 * sx + s2 * sy)$$

$$U_3(2,1) = c3 * sz - s3 * (c2 * sx + s2 * sy)$$

$$U_3(3,1) = s2 * sx - c2 * sy$$

$$U_3(1,2) = nz * s3 + c3 * (c2 * nx + ny * s2)$$

$$U_3(2,2) = c_3 * nz - s_3 * (c_2 * nx + ny * s_2)$$

$$U_3(3,2) = nx * s_2 - c_2 * ny$$

$$U_3(1,3) = az * s_3 + c_3 * (ax * c_2 + ay * s_2)$$

$$U_3(2,3) = az * c_3 - s_3 * (ax * c_2 + ay * s_2)$$

$$U_3(3,3) = ax * s_2 - ay * c_2$$

$$U_3(1,4) = s_3 * (Pz - r_1) + c_3 * (Px * c_2 + Py * s_2)$$

$$U_3(2,4) = c_3 * (Pz - r_1) - s_3 * (Px * c_2 + Py * s_2)$$

$$U_3(3,4) = Px * s_2 - Py * c_2$$

$$U_3(4,1) = U_3(4,2) = U_3(4,3) = 0$$

$$U_3(4,4) = 1$$

➤ Componentes de la matriz de la derecha 3T_7

$${}^3T_7 = \begin{bmatrix} c_4s_5s_7 - c_7(s_4s_6 - c_4c_5c_6) & s_7(s_4s_6 - c_4c_5c_6) + c_4s_5c_7 & s_4c_6 + c_4c_5s_6 & d_3 - s_4r_5 + r_7(s_4c_6 + c_4c_5s_6) \\ c_7(c_4s_6 + s_4c_5c_6) + s_4s_5s_7 & s_4s_5c_7 - s_7(c_4s_6 + s_4c_5c_6) & s_4c_5s_6 - c_4c_6 & c_4r_5 - r_7(c_4c_6 - s_4c_5s_6) \\ c_5s_7 - s_5c_6c_7 & c_5c_7 + s_5c_6s_7 & -s_5s_6 & -s_5s_6r_7 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

9.1.5 Iteración 5: $U_2 = {}^2T_7$

➤ Componentes de la matriz de la izquierda $U_2 = {}^2T_1 {}^1T_0 * U_0$

$$U_2(1,1) = c_2 * sx + s_2 * sy$$

$$U_2(2,1) = c_2 * sy - s_2 * sx$$

$$U_2(3,1) = sz$$

$$U_2(1,2) = c_2 * nx + ny * s_2$$

$$U_2(2,2) = c_2 * ny - nx * s_2$$

$$U_2(3,2) = nz$$

$$U_2(1,3) = ax * c_2 + ay * s_2$$

$$U_2(2,3) = ay * c_2 - ax * s_2$$

$$U_2(3,3) = az$$

$$U_2(1,4) = Px * c_2 + Py * s_2$$

$$U_2(2,4) = Py * c_2 - Px * s_2$$

$$U_2(3,4) = Pz - r_1$$

$$U_2(4,1) = U_2(4,2) = U_2(4,3) = 0$$

$$U_2(4,4) = 1$$

➤ Componentes de la matriz de la derecha 2T_7

$${}^2T_7(1,1) = s_5 * s_7 * (c_3 * c_4 - s_3 * s_4) - c_7 * (s_6 * (c_3 * s_4 + c_4 * s_3) - c_5 * c_6 * (c_3 * c_4 - s_3 * s_4))$$

$${}^2T_7(2,1) = c_2 * s_y - s_2 * s_x$$

$${}^2T_7(2,1) = c_6 * c_7 * s_5 - c_5 * s_7$$

$${}^2T_7(3,1) = c_7 * (s_6 * (c_3 * c_4 - s_3 * s_4) + c_5 * c_6 * (c_3 * s_4 + c_4 * s_3)) + s_5 * s_7 * (c_3 * s_4 + c_4 * s_3)$$

$${}^2T_7(1,2) = s_7 * (s_6 * (c_3 * s_4 + c_4 * s_3) - c_5 * c_6 * (c_3 * c_4 - s_3 * s_4)) + c_7 * s_5 * (c_3 * c_4 - s_3 * s_4)$$

$${}^2T_7(2,2) = -c_5 * c_7 - c_6 * s_5 * s_7$$

$${}^2T_7(3,2) = c_7 * s_5 * (c_3 * s_4 + c_4 * s_3) - s_7 * (s_6 * (c_3 * c_4 - s_3 * s_4) + c_5 * c_6 * (c_3 * s_4 + c_4 * s_3))$$

$${}^2T_7(1,3) = c_6 * (c_3 * s_4 + c_4 * s_3) + c_5 * s_6 * (c_3 * c_4 - s_3 * s_4)$$

$${}^2T_7(2,3) = s_5 * s_6$$

$${}^2T_7(3,3) = c_5 * s_6 * (c_3 * s_4 + c_4 * s_3) - c_6 * (c_3 * c_4 - s_3 * s_4)$$

$${}^2T_7(1,4) = c_3 * d_3 - r_5 * (c_3 * s_4 + c_4 * s_3) + r_7 * (c_6 * (c_3 * s_4 + c_4 * s_3) + c_5 * s_6 * (c_3 * c_4 - s_3 * s_4))$$

$${}^2T_7(2,4) = r_7 * s_5 * s_6$$

$${}^2T_7(3,4) = d_3 * s_3 + r_5 * (c_3 * c_4 - s_3 * s_4) - r_7 * (c_6 * (c_3 * c_4 - s_3 * s_4) - c_5 * s_6 * (c_3 * s_4 + c_4 * s_3))$$

$${}^2T_7(4,1) = {}^2T_7(4,2) = {}^2T_7(4,3) = 0$$

$${}^2T_7(4,4) = 1$$

9.1.6 Iteración 6: $U_1 = {}^1T_7$

➤ Componentes de la matriz de la izquierda $U_1 = {}^1T_0 * U_0$

$$U_0 = \begin{bmatrix} s_x & n_x & a_x & P_x \\ s_y & n_y & a_y & P_y \\ s_z & n_z & a_z & P_z - r_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

➤ Componentes de la matriz de la derecha 1T_7

$${}^1T_7(1,1) = s7 * (c5 * s2 + s5 * (c2 * c3 * c4 - c2 * s3 * s4)) - c7 * (s6 * (c2 * c3 * s4 + c2 * c4 * s3) + c6 * (s2 * s5 - c5 * (c2 * c3 * c4 - c2 * s3 * s4)))$$

$${}^1T_7(2,1) = -c7 * (s6 * (c3 * s2 * s4 + c4 * s2 * s3) - c6 * (c2 * s5 + c5 * (c3 * c4 * s2 - s2 * s3 * s4))) - s7 * (c2 * c5 - s5 * (c3 * c4 * s2 - s2 * s3 * s4))$$

$${}^1T_7(3,1) = c7 * (s6 * (c3 * c4 - s3 * s4) + c5 * c6 * (c3 * s4 + c4 * s3)) + s5 * s7 * (c3 * s4 + c4 * s3)$$

$${}^1T_7(1,2) = s7 * (s6 * (c2 * c3 * s4 + c2 * c4 * s3) + c6 * (s2 * s5 - c5 * (c2 * c3 * c4 - c2 * s3 * s4))) + c7 * (c5 * s2 + s5 * (c2 * c3 * c4 - c2 * s3 * s4))$$

$${}^1T_7(2,2) = s7 * (s6 * (c3 * s2 * s4 + c4 * s2 * s3) - c6 * (c2 * s5 + c5 * (c3 * c4 * s2 - s2 * s3 * s4))) - c7 * (c2 * c5 - s5 * (c3 * c4 * s2 - s2 * s3 * s4))$$

$${}^1T_7(3,2) = c7 * s5 * (c3 * s4 + c4 * s3) - s7 * (s6 * (c3 * c4 - s3 * s4) + c5 * c6 * (c3 * s4 + c4 * s3))$$

$${}^1T_7(1,3) = c6 * (c2 * c3 * s4 + c2 * c4 * s3) - s6 * (s2 * s5 - c5 * (c2 * c3 * c4 - c2 * s3 * s4))$$

$${}^1T_7(2,3) = c6 * (c3 * s2 * s4 + c4 * s2 * s3) + s6 * (c2 * s5 + c5 * (c3 * c4 * s2 - s2 * s3 * s4))$$

$${}^1T_7(3,3) = c5 * s6 * (c3 * s4 + c4 * s3) - c6 * (c3 * c4 - s3 * s4)$$

$${}^1T_7(1,4) = r7 * (c6 * (c2 * c3 * s4 + c2 * c4 * s3) - s6 * (s2 * s5 - c5 * (c2 * c3 * c4 - c2 * s3 * s4))) - r5 * (c2 * c3 * s4 + c2 * c4 * s3) + c2 * c3 * d3$$

$${}^1T_7(2,4) = r7 * (c6 * (c3 * s2 * s4 + c4 * s2 * s3) + s6 * (c2 * s5 + c5 * (c3 * c4 * s2 - s2 * s3 * s4))) - r5 * (c3 * s2 * s4 + c4 * s2 * s3) + c3 * d3 * s2$$

$${}^1T_7(3,4) = d3 * s3 + r5 * (c3 * c4 - s3 * s4) - r7 * (c6 * (c3 * c4 - s3 * s4) - c5 * s6 * (c3 * s4 + c4 * s3))$$

$${}^1T_7(4,1) = {}^1T_7(4,2) = {}^1T_7(4,3) = 0; \quad {}^1T_7(4,4) = 1$$

9.2 CODIGOS MATLAB

9.2.1 Modelamiento Geométrico Directo

9.2.1.1 Parámetros Khalil – Kleinfinger

9.2.1.1.1 Función MGD_Khalil

Función que nos da el modelamiento geométrico directo utilizando los parámetros Khalil-Kleinfinger, es decir, la posición y orientación final del laparoscopio.

```
% Modelamiento geométrico directo
% Asistente robótico Cirugía Laparoscópica 7GDL
% -----

% Parametros:
% a_j = angulo entre Z(j-1) y Zj alrededor de X(j-1)
% d_j = distancia entre Z(j-1) y Zj a lo largo de X(j-1)
% t_j = angulo entre X(j-1) y Xj alrededor de Zj
% r_j = distancia entre X(j-1) y Xj a lo largo de Zj

clear all
clc
syms t2 t3 t4 t5 t6 t7 r1 r7 d3 r5
% % Valores reales del IMEbot
%     r1=1.05; % base + art.1(asumimos cte) + art.2
%     d3=0.5; % brazo art.3
%     r5=0.4; % art.4 + art.5
%     r7=0.45; % art.6 + laparoscopio art.7

% MTH(a,d,t,r) matriz de transformación homogenea
% Nos devuelve la matriz directa e inversa
% T1==> frame que va de R0 a R1
% T1_inv ==> frame que va de R1 a R0
    [T1,T1_inv] = MTH(0,0,0,r1);
    [T2,T2_inv] = MTH(0,0,t2,0);
    [T3,T3_inv] = MTH(90,0,t3,0);
    [T4,T4_inv] = MTH(0,d3,t4,0);
    [T5,T5_inv] = MTH(-90,0,t5,r5);
    [T6,T6_inv] = MTH(90,0,t6,0);
    [T7,T7_inv] = MTH(90,0,t7,r7);

T01=factoriz_MGD(MULT(T1));
T02=factoriz_MGD(MULT(T1,T2));
T03=factoriz_MGD(MULT(T1,T2,T3));
T04=factoriz_MGD(MULT(T1,T2,T3,T4));
T05=factoriz_MGD(MULT(T1,T2,T3,T4,T5));
T06=factoriz_MGD(MULT(T1,T2,T3,T4,T5,T6));
T07=factoriz_MGD(MULT(T1,T2,T3,T4,T5,T6,T7)) % MGD

X=T07(1,4); Y=T07(2,4); Z=T07(3,4);
Sx=T07(1,1); Sy=T07(2,1); Sz=T07(3,1);
Nx=T07(1,2); Ny=T07(2,2); Nz=T07(3,2);
Ax=T07(1,3); Ay=T07(2,3); Az=T07(3,3);
```

9.2.1.1.2 Función MTH_Khalil

Función que calcula la matriz de transformación homogénea del sistema ${}^{j-1}T_j$ y ${}^jT_{j-1}$.

```
% Matriz de transformacion homogenea
% Parametros Khalil - Kleinfinger
% Definiendo el sistema Rj relativo a R(j-1)
% Modeling, identification and control of robots', pág. 38

function [T,T_inv] = MTH(a,d,t,r)
% alpha_j = angulo entre Z(j-1) y Zj alrededor de X(j-1)
% d_j      = distancia entre Z(j-1) y Zj a lo largo de X(j-1)
% theta_j  = angulo entre X(j-1) y Xj alrededor de Zj
% r_j      = distancia entre X(j-1) y Xj a lo largo de Zj
% T = Rot(X,alpha)*Trans(X,d)*Rot(Z,theta)*Trans(Z,r)

alpha = a*pi/180;
Ca = cos(alpha);
    if Ca <= 1e-15
        Ca=0; end
Sa = sin(alpha);

if isnumeric(t) == 1
    theta = t*pi/180;
    Ct = cos(theta);
    if Ct <= 1e-15 % Activar cuando son variables numéricas
        Ct=0; end
else
    theta = t;
    Ct = cos(theta);
end
St = sin(theta);

T = [ Ct      -St      0      d;
      Ca*St   Ca*Ct   -Sa   -r*Sa;
      Sa*St   Sa*Ct   Ca    r*Ca;
      0       0       0     1  ];

T_inv = [ Ct      Ca*St      Sa*St      -d*Ct;
          -St     Ca*Ct      Sa*Ct      d*St;
          0      -Sa        Ca         -1*r;
          0       0         0          1  ];
```

9.2.1.2 Parámetros Denavit – Hartenberg

9.2.1.2.1 Función MGD_DH

Función que nos da el MGD utilizando los parámetros Denavit- Hartenberg.

```

%=====
%//// DISEÑO, SIMULACIÓN Y CONTROL DE UN ASISTENTE ROBÓTICO ////
%////////// PARA CIRUGÍA LAPAROSCÓPICA //////////
%////////// Universidad de Piura //////////
%////////// Facultad de Ingeniería Mecánica - Eléctrica //////////
%////////// MANUEL AMAYA GONZALEZ //////////
%=====

% MODELAMIENTO GEOMÉTRICO DIRECTO
% Asistente robótico Cirugía Laparoscópica 7GDL
% -----

function [X,Y,Z]=MGD(t2,t3,t4,t5,t6,t7)

%% Valores reales de IMEbot
    D1=0.95; % base + eslabon 1(asumimos cte)
    L2=0.1; % eslabon2
    A3=0.5; % eslabon3
    L5=0.4; % eslabon4 + eslabon5
    L7=0.45; % eslabon6 + laparoscopia eslabon7

%% Valores de los ángulos
    aux=pi/180; % Conversión a radianes
    t2=t2*aux; % art.2
    t3=t3*aux; % art.3
    t4=t4*aux; % art.4
    t5=t5*aux; % art.5
    t6=t6*aux; % art.6
    t7=t7*aux; % art.7

% Matriz de transformación homogénea
% Nos devuelve la matriz directa e inversa
% T1==> frame que va de S0 a S1
% T1_inv ==> frame que va de S1 a S0

%
%           MTH(alpha, a,   theta,   d)
[T1,T1_inv] = MTH(  0,  0,      0,   D1);
[T2,T2_inv] = MTH(-90,  0,     t2,   L2);
[T3,T3_inv] = MTH(  0, A3, t3-pi/2,  0);
[T4,T4_inv] = MTH( 90,  0,  t4+pi,  0);
[T5,T5_inv] = MTH( 90,  0,  t5+pi,  L5);
[T6,T6_inv] = MTH( 90,  0, t6-pi/2,  0);
[T7,T7_inv] = MTH(  0,  0,     t7,  L7);

T00=eye(4);
T01=T1; T02=T01*T2; T03=T02*T3;
T04=T03*T4; T05=T04*T5; T06=T05*T6;
T07=T06*T7; % MGD

X=T07(1,4);
Y=T07(2,4);
Z=T07(3,4);

```

9.2.1.2.2 Función MTH_DH

Función que calcula la matriz de transformación homogénea del sistema..

```

%=====
%//// DISEÑO, SIMULACIÓN Y CONTROL DE UN ASISTENTE ROBÓTICO ////
%////////// PARA CIRUGÍA LAPAROSCÓPICA //////////
%////////// Universidad de Piura //////////
%////////// Facultad de Ingeniería Mecánica - Eléctrica //////////
%////////// MANUEL AMAYA GONZALEZ //////////
%=====

% Matriz de transformacion homogenea
% Parámetros Denavit-Hartenberg
% Relacionan el sistema {S_i-1} con el sistema a {S_i}
% "Fundamentos de Robótica", pág. 125

function [T,T_inv] = MTH(alpha,a,theta,d)
% alpha_i = angulo entre Z(i-1) y Z(i) alrededor de X(i)
% a_i      = distancia entre Z(i-1) y Z(i) a lo largo de X(i)
% theta_i  = angulo entre X(i-1) y X(i) alrededor de Z(i-1)
% d_i      = distancia entre X(i-1) y X(i) a lo largo de Z(i-1)
% (i-1)_A_i = [Rot(Z(i-1),theta_i)]*[Trans(Z(i-1),d_i)]*...
%              [Trans(X(i),a_i)]*[Rot(X(i),alpha_i)]

alpha = alpha*pi/180; %alpha ingresa en °S
Ca = Redondear(cos(alpha));
Sa = Redondear(sin(alpha));

if isnumeric(theta) == 1
    Ct = Redondear(cos(theta));
    St = Redondear(sin(theta));
else
    Ct = cos(theta);
    St = sin(theta);
end

T = [ Ct   -Ca*St   Sa*St   a*Ct;
      St    Ca*Ct  -Sa*Ct   a*St;
      0     Sa     Ca      d  ;
      0     0     0       1  ];

T_inv = [   Ct      St      0   -a;
           -Ca*St   Ca*Ct   Sa  -d*Sa;
           Sa*St   -Sa*Ct   Ca  -d*Ca;
           0       0       0    1  ];
if isnumeric(theta) == 0
    T=simplify(T);
    T_inv=simplify(T_inv);
end

```

9.2.2 Modelamiento Geométrico Inverso

9.2.2.1 Parámetros Khalil – Kleinfinger

9.2.2.1.1 Función MGI_Khalil

Calcula el modelamiento geométrico inverso utilizando los parámetros Khalil-Kleinfinger y el método numérico de la bisección para hallar θ_3 . Además se implementa la función 'VNum' que permite obtener el valor numérico de las ecuaciones simbólicas.

```
% Modelamiento geométrico inverso
% Asistente robótico Cirugía Laparoscópica 7GDL
% -----

% Ejecutar primero el archivo MGD.m
syms sx nx ax Px sy ny ay Py sz nz az Pz
syms c2 c3 c4 c5 c6 c7 s2 s3 s4 s5 s6 s7 c34 s34

% Términos del Método de Paul
% -----
U0 = [ sx  nx  ax  Px;    % Valores conocidos
      sy  ny  ay  Py
      sz  nz  az  Pz
      0   0   0   1  ];
U1=MULT(T1_inv,U0); T17=MULT(T2,T3,T4,T5,T6,T7);
U2=MULT(T2_inv,U1); T27=MULT(T3,T4,T5,T6,T7);
U3=MULT(T3_inv,U2); T37=MULT(T4,T5,T6,T7);
U4=MULT(T4_inv,U3); T47=MULT(T5,T6,T7);
U5=MULT(T5_inv,U4); T57=MULT(T6,T7);
U6=MULT(T6_inv,U5); T67=MULT(T7);

% Iteración 5: Cálculo theta2
% -----
a1=(r7*ax-Px);
b1=(Py-r7*ay);
theta2=atan2(b1,-a1);
theta2_1=atan2(-b1,a1); % Ecuación 3.42
theta2=VNum(theta2)*180/pi;
theta2=vpa(theta2);
theta2=double(theta2)

% Ecuación 3.18: Calculo theta4.
% -----
syms tx ty tz
Pt=[tx;ty;tz];
P06=T06(1:3,4); % Posicion 6
P07=U0(1:3,4); % Conocido
mod=(P07-Pt); mod=mod.^2;
K2=r7-sqrt(mod(1,1)+mod(2,1)+mod(3,1));K2=K2^2; % Término K2

mod2=(Pt-P06); mod2=mod2.^2;
K1=mod2(1,1)+mod2(2,1)+mod2(3,1);
K1=collect(K1,'c34');K1=collect(K1,'s34');
K1=subs(K1,'c34',c3*c4 - s3*s4);
K1=subs(K1,'s34',c3*s4 + c4*s3);
K1=collect(K1,'r5');K1=subs(K1,'1.0*c2^2 + 1.0*s2^2',1);
K1=collect(K1,'c4');K1=collect(K1,'s4');
```

```

K1=subs(K1,'c3^2 + s3^2',1);
K1=collect(K1,'c4');K1=collect(K1,'s4');
K1=collect(K1,'r5^2');K1=subs(K1,'c4^2 + s4^2',1);
K1=collect(K1,'c4');K1=collect(K1,'s4'); % Término K1
% Dando la forma: K1=A*s4+B*c4+C = K2
A=r5*(c3*(2.0*c2*(tx - c2*c3*d3) + 2.0*s2*(ty - c3*d3*s2)) - 2*s3*(r1 -
tz + d3*s3));
B=r5*(s3*(2.0*c2*(tx - c2*c3*d3) + 2.0*s2*(ty - c3*d3*s2)) + 2*c3*(r1 -
tz + d3*s3));
C=(r1 - tz + d3*s3)^2 + (tx - c2*c3*d3)^2 + (ty - c3*d3*s2)^2 + r5^2;

% Dando la forma: a*sin(theta)+b*cos(theta)=c
% theta= atan2(a,b)-atan2(sqrt(a^2+b^2-c^2),c)
% theta= atan2(a,b)+atan2(sqrt(a^2+b^2-c^2),c)
a=vpa(A); b=vpa(B); c=vpa(K2-C);
Aux1=a*s4+b*c4-c; %Ecuación 3.40
theta4=atan2(a,b)-atan2(sqrt(a^2+b^2-c^2),c);
theta4_1=atan2(a,b)+atan2(sqrt(a^2+b^2-c^2),c);

% Iteración 4 y 6: Cálculo theta3
% -----
Aux2=Pz-r1-r7*az-d3*s3-c34*r5; %Ecuacion 3.43=0
Aux3=c3*(Pz-r1-r7*az)+s3*(-Px*c2-Py*s2+r7*ax*c2+r7*ay*s2) -
c4*r5; %Ecuacion 3.44=0
R=-c3*s3*r5; S=-r5*s3^2; Q=s3*(Px*c2+Py*s2-r7*ax*c2-r7*ay*s2-c3*d3);
Aux4=R*s4+S*c4-Q; %Ecuación 3.45=0
% Dando la forma: R*s4+S*c4= Q
R=vpa(R); S=vpa(S); Q=vpa(Q);
theta4x=atan2(R,S)-atan2(sqrt(R^2+S^2-Q^2),Q);
theta4x_1=atan2(R,S)+atan2(sqrt(R^2+S^2-Q^2),Q);

% % METODO NUMERICO: THETA3
theta3=Biseccion(theta4,theta4x,theta2,10,115,0.0001)
theta4=VNum(theta4,theta2,theta3);
theta4=vpa(theta4*180/pi);
theta4=double(theta4)

% Iteración 3: Cálculo theta5
% -----
az4=U4(3,3);
ax4=U4(1,3);
theta5=atan2(-az4,ax4);
theta5=VNum(theta5,theta2,theta3,theta4);
theta5=vpa(theta5*180/pi);
theta5=double(theta5)

% Cálculo de theta6.
% -----
% Restriccion del trócar (Límites)
P07=T07(1:3,4); % Posición 7 ó P7=[Px;Py;Pz];
Lim=cross((Pt-P06),(P07-Pt)); % Producto cruz
Lim=collect(Lim,'c6');Lim=collect(Lim,'s6');
M1=((c2*r7*s5 + c5*c34*r7*s2)*(r1 - tz + c34*r5 + d3*s3) + c5*r7*s34*(ty
- c3*d3*s2 + r5*s2*s34));
N1=(r7*s2*s34*(r1 - tz + c34*r5 + d3*s3) - c34*r7*(ty - c3*d3*s2 +
r5*s2*s34));
M2=((r7*s2*s5 - c2*c5*c34*r7)*(r1 - tz + c34*r5 + d3*s3) - c5*r7*s34*(tx
- c2*c3*d3 + c2*r5*s34));
N2=(c34*r7*(tx - c2*c3*d3 + c2*r5*s34) - c2*r7*s34*(r1 - tz + c34*r5 +
d3*s3));

```

```

theta6=atan2((N2-N1),(M1-M2));
theta6=VNum(theta6,theta2,theta3,theta4,theta5);
theta6=vpa(theta6*180/pi);
theta6=double(theta6)
% Iteración 1: Cálculo de theta7.
% -----
sz6=U6(3,1);
sx6=U6(1,1);
theta7=atan2(sz6,sx6); %range [-?/2, ?/2]
theta7=VNum(theta7,theta2,theta3,theta4,theta5,theta6);
theta7=vpa(theta7*180/pi);
theta7=double(theta7)

Pos_final=VNum(T07,theta2,theta3,theta4,theta5,theta6,theta7)

```

9.2.2.1.2 Método de la Bisección

Método numérico iterativo implementado para el cálculo de θ_3 .

```

% Método de la bisección
function [xr]=Biseccion(theta4,theta4x,theta2,xa,xb,tol)

f1a= vpa(VNum(theta4,theta2,xa)*180/pi);
f2a= vpa(VNum(theta4x,theta2,xa)*180/pi);
if f2a>=95
    f2a=f2a-360;
end
fxa=f1a-f2a;

f1b= vpa(VNum(theta4,theta2,xb)*180/pi);
f2b= vpa(VNum(theta4x,theta2,xb)*180/pi);
if f2b>=95
    f2b=f2b-360;
end
fxb=f1b-f2b;

if sign(fxa)==sign(fxb)
    disp('No tiene solución por método de la bisección')
    return
end
while (xb-xa)/2>tol
    xr=(xa+xb)/2;
    f1r= vpa(VNum(theta4,theta2,xr)*180/pi);
    f2r= vpa(VNum(theta4x,theta2,xr)*180/pi);
    if f2r>=95
        f2r=f2r-360;
    end
    fxr=f1r-f2r;
    if sign(fxa)==sign(fxr)
        xa=xr;
        fxa=fxr;
    else
        xb=xr;
        fxb=fxr;
    end
end
end

```

9.2.2.2 Parámetros Denavit - Hartenberg

9.2.2.2.1 Función MGI_DH

Calcula el modelamiento geométrico inverso utilizando los parámetros Denavit-Hartenberg.

```

%=====
%//// DISEÑO, SIMULACIÓN Y CONTROL DE UN ASISTENTE ROBÓTICO ////
%////////// PARA CIRUGÍA LAPAROSCÓPICA ////////////
%////////// Universidad de Piura ////////////
%////////// Facultad de Ingeniería Mecánica - Eléctrica ////////////
%////////// MANUEL AMAYA GONZALEZ ////////////
%=====

% MODELAMIENTO GEOMÉTRICO INVERSO
% Asistente robótico Cirugía Laparoscópica 7GDL
% -----

% Ejecutar primero el archivo MGD.m
function [q2,q3,q4,q5,q6,R]=MGI(Px,Py,Pz,theta7)
tic
syms ox nx ax oy ny ay oz nz az
syms c2 c3 c4 c5 c6 c7 s2 s3 s4 s5 s6 s7 c34 s34
% clc
%% Posicion final deseada y trócar
% Pf=[0.5;0;0.8]; %Posición final
Pf=[Px;Py;Pz]; %Posición final
% Pt=[tx;ty;tz]; %Posición del trócar
Pt=[0.5;0;1]; %Posición del trócar fijo

%% Orientación de Z7
Z7=(Pf-Pt)/norm((Pf-Pt),2);
U0 = [ nx  ox  Z7(1,1)  Px;      % Valores conocidos
       ny  oy  Z7(2,1)  Py;
       nz  oz  Z7(3,1)  Pz;
       0   0   0   1  ];
% U0 = subs(U0,'ax',Z7(1,1));
% U0 = subs(U0,'ay',Z7(2,1));
% U0 = subs(U0,'az',Z7(3,1));

%% Desacoplo Cinemático - Método geométrico
% Valores reales de IMEbot
% =====
D1=0.95; % base + eslabon 1(asumimos cte)
L2=0.1; % eslabon2
A3=0.5; % eslabon3
L5=0.4; % eslabon4 + eslabon5
L7=0.45; % eslabon6 + laparoscopia eslabon7

%% Cálculo theta2
% =====
Pm=Pf-L7*Z7;
Xm=Pm(1,1);Ym=Pm(2,1);Zm=Pm(3,1);
theta2=atan2(Ym,Xm); % t2 en radianes
theta2=theta2*180/pi;
q2=theta2;

%% Cálculo theta4
% =====

```

```

cp4=(Xm^2+Ym^2+(Zm-D1-L2)^2-A3^2-L5^2)/(2*A3*L5);
sp4=sqrt(1-cp4^2);% sp4_1=-sqrt(1-cp4^2);
phi4=atan2(sp4,cp4);% phi4_1=atan2(sp4_1,cp4);
Phi4=phi4;
theta4=Phi4-pi/2;

%% Cálculo theta3
% =====
a=atan2(L5*sin(Phi4),A3+L5*cos(Phi4));
b=atan2((Zm-D1-L2),sqrt(Xm^2+Ym^2));
% b1=atan2((Zm-D1-L2),-sqrt(Xm^2+Ym^2));
Phi3=a+b;
theta3=pi/2-Phi3;
theta3=theta3*180/pi;
theta4=theta4*180/pi;
q3=-90+theta3;
q4=180+theta4;

% Valores necesarios para el cálculo inicial
% Se 'comentan' para evitar la carga computacional
% R1=T1(1:3,1:3);% R1_inv=T1_inv(1:3,1:3);
% R2=T2(1:3,1:3);% R2_inv=T2_inv(1:3,1:3);
% R3=T3(1:3,1:3);% R3_inv=T3_inv(1:3,1:3);
% R4=T4(1:3,1:3);% R4_inv=T4_inv(1:3,1:3);
% R5=T5(1:3,1:3);
% R6=T6(1:3,1:3);
% R7=T7(1:3,1:3);
% R04=Reducir(R1*R2*R3*R4);
% R04_inv=Reducir(R4_inv*R3_inv*R2_inv*R1_inv)
% R47=Reducir(R5*R6*R7);

%% Cálculo theta5 y 6
% =====
t34=theta3+theta4;
si2=sind(theta2); co2=cosd(theta2);
si34=sind(t34);co34=cosd(t34);
w1=si2*si34; w2=co2*si34; w3=co2*co34; w4=si2*co34;
R04_inv=[ -w2, -w1,-co34;
          -si2, co2, 0;
          w3, w4,-si34];
Aux=R04_inv*U0(1:3,1:3);
Aux=vpa(Aux);
Aux_13=Aux(1,3);
Aux_23=Aux(2,3);
Aux_33=Aux(3,3);
theta5=atan2(Aux_23,Aux_13);
theta6=asin(-Aux_33);
theta5=double(theta5*180/pi);
if theta5==360
    theta5=0;
end
theta6=double(vpa(theta6*180/pi));
q5=double(180+theta5);
q6=double(-90+theta6);

%% De MGD_sym.m
si5=sind(theta5); co5=cosd(theta5);
si6=sind(theta6); co6=cosd(theta6);
si7=sind(theta7); co7=cosd(theta7);

```

```

j1=si5*si7; j2=co5*si7; j3=co5*co7; j4=si5*co7;
j5=co6*co7; j7=co6*si7;
u1=j1+j3*si6; % (s5*s7 + c5*c7*s6)
u2=j2-j4*si6; % (c5*s7 - s5*c7*s6)
u3=j4-j2*si6; % (s5*c7 - c5*s7*s6)
u4=j3+j1*si6; % (c5*c7 + s5*s7*s6)

Nx = w2*u1 - si2*u2 - j5*w3; %Nx = c2*s34*(s5*s7 + c5*c7*s6) -
s2*(c5*s7 - s5*c7*s6) - c6*c7*c2*c34
Ny = w1*u1 + co2*u2 - j5*w4; %Ny = s2*s34*(s5*s7 + c5*c7*s6) +
c2*(c5*s7 - s5*c7*s6) - c6*c7*s2*c34
Nz =co34*u1 + j5*si34;%Nz = c34*(s5*s7 + c5*c7*s6) +
c6*c7*s34
Ox = w2*u3 - si2*u4 + j7*w3; %Ox = c2*s34*(s5*c7 - c5*s7*s6) -
s2*(c5*c7 + s5*s7*s6) + c6*s7*c2*c34
Oy = w1*u3 + co2*u4 + j7*w4; %Oy = s2*s34*(s5*c7 - c5*s7*s6) +
c2*(c5*c7 + s5*s7*s6) + c6*s7*s2*c34
Oz =co34*u3 - j7*si34;%Oz = c34*(s5*c7 - c5*s7*s6) -
c6*s7*s34

R = [ Nx Ox Z7(1,1); % Matriz de rotación
      Ny Oy Z7(2,1);
      Nz Oz Z7(3,1)];

toc

% theta7=0; %Manual, regulado por el médico
% q7=theta7

```

9.2.3 Trayectoria Esférica

```

%% TRAYECTORIA 1: ESFERA
% Permite crear la trayectoria de una esfera
% Ejecutar este script antes que todo.

function Esferal = T1_Esferal()
clc
% Coordenadas polares
xc=0.5;
yc=0;
zc=0.9;
r=0.05; %radio de la circunferencia
% x=xc+r*sin(theta)*cos(phi);
% y=yc+r*sin(theta)*sin(phi);
% z=zc+r*cos(phi);

m = 20; % numero de circunferencias
n = 32; % maximo nro de puntos en una circunferencia
i = 1;
theta = 0;
for j = 1 : 10
    for phi = linspace(0, 2*pi, 3*j-1)
        Pos(:, :, i) = [xc+r*sin(theta)*cos(phi);
                        yc+r*sin(theta)*sin(phi);
                        zc+r*cos(theta)];
        i = i+1;
    end;
    theta = theta + pi / m;
end;

```

```

for j = 11:-1:1
    for phi = linspace(0, 2*pi, 3*j-1)
        Pos(:, :, i) = [xc+r*sin(theta)*cos(phi);
                        yc+r*sin(theta)*sin(phi);
                        zc+r*cos(theta)];
        i = i+1;
    end;
    theta = theta + pi / m;
end;
% M=Pos(:, :, 1)
dato= size(Pos,3);
Pos = Pos(:, :, 2:(dato-1));
tam = size(Pos,3);
for i=1:tam
    P(i, 1) = Redondear2(Pos(1, 1, i));
    P(i, 2) = Redondear2(Pos(2, 1, i));
    P(i, 3) = Redondear2(Pos(3, 1, i));
    % plot3(P(:, 1), P(:, 2), P(:, 3), 'b');
    % hold off, axis equal;
    % view(3), xlabel('x'), ylabel('y'), zlabel('z'), grid on;
    % pause(.00001)
end

t=linspace(1,tam,tam); % Cada tramo en 1 segundo
t=t';
Esferal.time=t;
Esferal.signals.values = [P(:, 1), P(:, 2), P(:, 3)];
Esferal.signals.dimensions =3;

```

9.2.4 Generación de Trayectoria

9.2.4.1 Función Interpolar

```

%% INTERPOLACIÓN: ESFERA
% % Permite crear una función 'continua' con una trayectoria suave y
% utilizando la velocidad calculada en el Jacobiano.
% -----

function [Qt,Qpt,Qppt] = Interpolar(NQ,NQp)

q2 = NQ(:,1); q3 = NQ(:,2); q4 = NQ(:,3);
q5 = NQ(:,4); q6 = NQ(:,5); q7 = NQ(:,6);
qp2 = NQp(:,1); qp3 = NQp(:,2); qp4 = NQp(:,3);
qp5 = NQp(:,4); qp6 = NQp(:,5); qp7 = NQp(:,6);
[t1,t2]=size(NQ);
t=linspace(1,t1,t1)';
n=10; % nro de divisiones del Ts

Path2=splin_cubico1(q2,qp2,t,n);
Path3=splin_cubico1(q3,qp3,t,n);
Path4=splin_cubico1(q4,qp4,t,n);
Path5=splin_cubico1(q5,qp5,t,n);
Path6=splin_cubico1(q6,qp6,t,n);
Path7=splin_cubico1(q7,qp7,t,n);
[m,m1]=size(Path2);
% Path1=zeros(m,m1);
TQ(:, :) = [Path2(:,1) Path3(:,1) Path4(:,1) Path5(:,1) Path6(:,1)
Path7(:,1)];%Path1(:,1)

```

```

TQp(:, :) = [Path2(:,2) Path3(:,2) Path4(:,2) Path5(:,2) Path6(:,2)
Path7(:,2)];
TQpp(:, :) = [Path2(:,3) Path3(:,3) Path4(:,3) Path5(:,3) Path6(:,3)
Path7(:,3)];
t2 = linspace(1/n, 340, m)';
Qt.time=t2;
Qt.signals.values = TQ;
Qt.signals.dimensions =6;
Qpt.time=t2;
Qpt.signals.values = TQp;
Qpt.signals.dimensions =6;
Qppt.time=t2;
Qppt.signals.values = TQpp;
Qppt.signals.dimensions =6;

```

9.2.4.2 Función splin_cúbico

```

%% Función Splin cúbico
% Obtiene los coeficientes que interpolan los
% valores q con las velocidades qd en los instantes t
%  $q(t) = a+b*(t-ti)+c*(t-ti)^2+d*(t-ti)^3$ 

function [Path]= splin_cubico1(q,qd,t,s)
% q es un vector de posiciones angulares
% qd es un vector de velocidades angulares
% t
[n m] = size(t);
j=1;
for i=1:n-1
    ti(i,1) = t(i,1);
    tii(i,1)= t(i+1,1);
    T      = tii(i,1)-ti(i,1);
    a(i,1)=  q(i,1);
    b(i,1)=  qd(i,1);
    c(i,1)=  (3/T^2)*(q(i+1,1)-q(i,1))- (1/T^2)*(qd(i+1,1)+2*qd(i,1));
    d(i,1)=  -(2/T^3)*(q(i+1,1)-q(i,1))+(1/T^2)*(qd(i+1,1)+ qd(i,1));

    % Asumiremos que el período de muestreo es el mismo
    % que el considerado en la generación de la esfera
    inc=T/s;
    for tt=ti(i,1):inc:(tii(i,1)-inc) %-inc
        Path(j,1) = a(i,1)+b(i,1)*(tt-ti(i,1))+c(i,1)*(tt-
ti(i,1))^2+d(i,1)*(tt-ti(i,1))^3; %posicion
        Path(j,2) = b(i,1)+2*c(i,1)*(tt-ti(i,1))+3*d(i,1)*(tt-
ti(i,1))^2;%velocidad
        Path(j,3) = 2*c(i,1)+ 6*d(i,1)*(tt-ti(i,1));%aceleracion
        j=j+1;
    end
end

T1      = 1;
ti1 = t(n,1);
a1=  q(n,1);
b1=  qd(n,1);
c1=  -(1/T1^2)*(2*qd(n,1));
d1=  (1/T1^2)*(qd(n,1));

```

```

for tt=ti1:inc:(ti1+T-inc) %-inc
    Path(j,1) = a1+ b1*(tt-ti1)+c1*(tt-ti1)^2+d1*(tt-ti1)^3;
    Path(j,2) = b1+2*c1*(tt-ti1)+3*d1*(tt-ti1)^2;
    Path(j,3) = 2*c1+ 6*d1*(tt-ti1);
    j=j+1;
end

```

9.2.5 Modelamiento cinemático

```

%% MODELAMIENTO CINEMÁTICO DIRECTO
% Asistente robótico Cirugía Laparoscópica 7GDL
% -----
% Relaciona las velocidades en las articulaciones con las velocidades
% en el espacio de trabajo.

function Vel = MCD(NQ,NQp)
t2=NQ(1,1); t3=NQ(2,1); t4=NQ(3,1); t5=NQ(4,1); t6=NQ(5,1); t7=NQ(6,1);
aux=pi/180; % Conversión a radianes
NQp=NQp*aux;
% tp2=NQp(1,1); tp3=NQp(2,1); tp4=NQp(3,1); tp5=NQp(4,1); tp6=NQp(5,1);
tp7=NQp(6,1);
%% Valores reales de IMEbot
    D1=0.95; % base + eslabon 1(asumimos cte)
    L2=0.1; % eslabon2
    A3=0.5; % eslabon3
    L5=0.4; % eslabon4 + eslabon5
    L7=0.45; % eslabon6 + laparoscopia eslabon7

%% Valores de los ángulos
    t2=t2*aux; % art.2
    t3=t3*aux; % art.3
    t4=t4*aux; % art.4
    t5=t5*aux; % art.5
    t6=t6*aux; % art.6
    t7=t7*aux; % art.7

%% MTH(alpha,a,theta,d) matriz de transformación homogenea
% Nos devuelve la matriz directa e inversa
% T1==> frame que va de S0 a S1
% T1_inv ==> frame que va de S1 a S0
    [T1,T1_inv] = MTH( 0, 0, 0, D1);
    [T2,T2_inv] = MTH( -90, 0, t2, L2);
    [T3,T3_inv] = MTH( 0, A3, t3-pi/2, 0);
    [T4,T4_inv] = MTH( 90, 0, t4+pi, 0);
    [T5,T5_inv] = MTH( 90, 0, t5+pi, L5);
    [T6,T6_inv] = MTH( 90, 0, t6-pi/2, 0);
    [T7,T7_inv] = MTH( 0, 0, t7, L7);

T00=eye(4);
T01=T1; T02=T01*T2; T03=T02*T3;
T04=T03*T4; T05=T04*T5; T06=T05*T6;
T07=T06*T7; % MGD

%% Jacobiano
z00=T00(1:3,3);
z01=T01(1:3,3);
z02=T02(1:3,3);
z03=T03(1:3,3);

```

```

z04=T04(1:3,3);
z05=T05(1:3,3);
z06=T06(1:3,3);
z07=T07(1:3,3);
p07=T07(1:3,4)-T00(1:3,4);
p17=T07(1:3,4)-T01(1:3,4);
p27=T07(1:3,4)-T02(1:3,4);
p37=T07(1:3,4)-T03(1:3,4);
p47=T07(1:3,4)-T04(1:3,4);
p57=T07(1:3,4)-T05(1:3,4);
p67=T07(1:3,4)-T06(1:3,4);

J1=[z00;0;0;0];
J2=[cross(z01,p17);z01];
J3=[cross(z02,p27);z02];
J4=[cross(z03,p37);z03];
J5=[cross(z04,p47);z04];
J6=[cross(z05,p57);z05];
J7=[cross(z06,p67);z06];

J=[J2 J3 J4 J5 J6 J7];
Vel=J*NQp; % Velocidad en m/s y rad/s
Vel=[Vel(1:3,1);Vel(4:6,1)/aux]; % Velocidad en m/s y °/s

```

9.2.6 Modelamiento Dinámico Inverso

9.2.6.1 Formulación Lagrangiana

```

%% MODELAMIENTO DINÁMICO INVERSO
% Mecánica Lagrangiana
% -----

function Tau = MDI_LG(q,qp,qpp)
%% Condiciones Iniciales
% Velocidades angulares a partir de la posición y el Jacobiano
q1=0; q2=q(1,1); q3=q(2,1); q4=q(3,1); q5=q(4,1); q6=q(5,1); q7=0;
[T1,T2,T3,T4,T5,T6,T7] = MGD_mod(q2,q3,q4,q5,q6);
qp1=0; qp2=qp(1,1)*pi/180; qp3=qp(2,1)*pi/180;
qp4=qp(3,1)*pi/180; qp5=qp(4,1)*pi/180; qp6=qp(5,1)*pi/180; qp7=0;
qpp1=0; qpp2=qpp(1,1)*pi/180; qpp3=qpp(2,1)*pi/180;
qpp4=qpp(3,1)*pi/180; qpp5=qpp(4,1)*pi/180; qpp6=qpp(5,1)*pi/180; qpp7=0;

%% L1 Asignar a cada eslabón un sistema {Si}

%% L2 Obtener las matrices de transformación T para cada elemento
% Se obtuvieron los valores previamente en la función 'Jacobiano'
T=zeros(4,4,7);
T(:,:,1) = T1;
T(:,:,2) = T2;
T(:,:,3) = T3;
T(:,:,4) = T4;
T(:,:,5) = T5;
T(:,:,6) = T6;
T(:,:,7) = T7;

A00=eye(4);
A=zeros(4,4,77);

```

```

for i=0:7
    Mul=eye(4);
    for j=1:7
        if i<=j
            if i==j
                A(:,:,10*i+j)= eye(4);
            else
                A(:,:,10*i+j)= Mul*T(:,:,j);
                Mul = A(:,:,10*i+j);
            end
        end
    end
end

%% L3 Obtener las matrices Uij
% Uij=dT/dqj=(T_0_j-1)*(Qj)*(T_j-1_i )
% i: indice de la MTH. Ejem: T0i
% j: indice de la variable articular. Ejem: qj
Qr=[0 -1 0 0;
    1 0 0 0;
    0 0 0 0;
    0 0 0 0];
Qt=[0 0 0 0;
    0 0 0 0;
    0 0 0 1;
    0 0 0 0];

Q=zeros(4,4,7);
for i=1:7
    if i==1
        Q(:,:,i) =Qt;
    else
        Q(:,:,i) =Qr;
    end
end

U=zeros(4,4,777);
for i=1:7
    for j=1:7
        if j<=i
            if j==1
                U(:,:,10*i+1)= A00*Q(:,:,j)*A(:,:,10*(j-1)+i);
            else
                U(:,:,10*i+j)= A(:,:, (j-1))*Qr*A(:,:,10*(j-1)+i);
                U(:,:,10*i+j)= A(:,:, (j-1))*Q(:,:,j)*A(:,:,10*(j-1)+i);
            end
        end
    end
end

%% L4 Obtener las matrices Uijk
% Uijk=dUij/dqk
for i=1:7
    for j=1:7
        for k=1:7
            if i>=k && k>=j
                if j==1
                    if k==1
                        U(:,:,100*i+10*j+k)=A00*Q(:,:,j)*A00*Q(:,:,k)*A(:,:,10*(k-1)+i);
                    else

```

```

        U(:, :, 100*i+10*j+k)=A00*Q(:, :, j)*A(:, :, 10*(j-1)+k-
1)*Q(:, :, k)*A(:, :, 10*(k-1)+i);
        end
        else
            U(:, :, 100*i+10*j+k)=A(:, :, j-1)*Q(:, :, j)*A(:, :, 10*(j-1)+k-
1)*Q(:, :, k)*A(:, :, 10*(k-1)+i);
            end
        end
        if i>=j && j>=k
            if k==1
                if j==1
                    U(:, :, 100*i+10*j+k)=A00*Q(:, :, k)*A00*Q(:, :, j)*A(:, :, 10*(j-1)+i);
                else
                    U(:, :, 100*i+10*j+k)=A00*Q(:, :, k)*A(:, :, 10*(k-1)+j-
1)*Q(:, :, j)*A(:, :, 10*(j-1)+i);
                end
            else
                U(:, :, 100*i+10*j+k)=A(:, :, k-1)*Q(:, :, k)*A(:, :, 10*(k-1)+j-
1)*Q(:, :, j)*A(:, :, 10*(j-1)+i);
            end
        end
    end
end
end
end

```

```
%% L5 Matrices de Pseudo-inercias
```

```
% sii=[SCx_i;SCy_i;SCz_i]
```

```
% Centro de masa del eslabón i respecto al sistema {S_i}
```

```
% Datos extraídos del las propiedades físicas de las piezas en
Solidworks
```

```
r=zeros(4,1,7);
```

```
r(:,1,1) = [ 0; 0; -0.1380; 1];
```

```
r(:,1,2) = [ 0; 0.0473; 0; 1];
```

```
r(:,1,3) = [-0.2994; 0; 0; 1];
```

```
r(:,1,4) = [ 0; 0; 0.0377; 1];
```

```
r(:,1,5) = [ 0; -0.1179; 0; 1];
```

```
r(:,1,6) = [ 0; 0; 0.0273; 1];
```

```
r(:,1,7) = [ 0; 0; -0.3116; 1];
```

```
% mii= Masa del eslabón i [kg]
```

```
M=zeros(7,1);
```

```
M(1,1) = 5.9664;
```

```
M(2,1) = 1.7791;
```

```
M(3,1) = 3.5773;
```

```
M(4,1) = 1.9849;
```

```
M(5,1) = 1.3959;
```

```
M(6,1) = 0.1200;
```

```
M(7,1) = 0.7399;
```

```
% Iii=[Ixx Ixy Ixz; Iyx Iyy Iyz; Izx Izy Izz]
```

```
% Matriz de inercia del eslabón i expresado en un sistema paralelo a
{S_i} y con el origen en el centro de masas del eslabón.
```

```
% Datos extraídos del las propiedades físicas de las piezas en
Solidworks
```

```
% que serán los valores de las inercias L.
```

```
I=zeros(3,3,7);
```

```
I(:, :, 1) = [ 0.2275 0 0 ;
```

```
0 0.2275 0 ;
```

```
0 0 0.0217 ];
```

```

I(:, :, 2) = [ 0.0082      0      0 ;
              0  0.0021      0 ;
              0      0  0.0086 ];

I(:, :, 3) = [ 0.0070      0      0 ;
              0  0.4152      0 ;
              0      0  0.4120 ];

I(:, :, 4) = [ 0.0075      0      0 ;
              0  0.0076      0 ;
              0      0  0.0016 ];

I(:, :, 5) = [ 0.0306      0      0 ;
              0  0.0006      0 ;
              0      0  0.0305 ];

I(:, :, 6) = [ 0.0002      0      0 ;
              0  0.0002      0 ;
              0      0  0.00001166 ];

I(:, :, 7) = [ 0.0749      0      0 ;
              0  0.0749      0 ;
              0      0  0.0001 ];

B=zeros(3,3,7);
J=zeros(4,4,7);
for i=1:7
    B(:, :, i)= I(:, :, i);
    B(1,1,i)=(-I(1,1,i) + I(2,2,i) + I(3,3,i))/2;
    B(2,2,i)=( I(1,1,i) - I(2,2,i) + I(3,3,i))/2;
    B(3,3,i)=( I(1,1,i) + I(2,2,i) - I(3,3,i))/2;
%
J(:, :, i)=cat(1, cat(2, B(:, :, i), M(i,1)*r(1:3,1,i)), cat(2, M(i,1)*r(1:3,1,i) .
', M(i,1)));
    J(:, :, i)=[B(:, :, i) M(i,1)*r(1:3,1,i);
               M(i,1)*r(1:3,1,i) .' M(i,1)];
end

%% L6 Obtener la matriz de inercias D=[dij]
D=zeros(7);
for i=1:7
    for k=1:7
        t = max([i k]);
        for j=t:7
            D(i,k)=D(i,k)+
trace(U(:, :, 10*j+k)*J(:, :, j)*(U(:, :, 10*j+i) .'));
        end
    end
end

%%Codigo resumido
qppx=[qpp1; qpp2; qpp3; qpp4; qpp5; qpp6; qpp7];
qpX=[qp1; qp2; qp3; qp4; qp5; qp6; qp7];
g=[0 0 -9.81 0];
D1=zeros(7,1);
H1=zeros(7,1);
C1=zeros(7,1);
for i=1:7
    for j=i:7
        C1(i)= C1(i)-M(j,1)*g*U(:, :, 10*j+i)*r(:, 1, j);
    end
end

```

```

        for k=1:j
            D1(i)=D1(i)+
trace(U(:,:,10*j+k)*J(:,:,j)*(U(:,:,10*j+i)).')*qpx(k,1);
            for m=1:j
                H1(i)=H1(i)+
trace(U(:,:,10*j+10*k+m)*J(:,:,j)*(U(:,:,10*j+i)).')*qpx(k,1)*qpx(m,1);
            end
        end
    end
end

%% L7 Obtener los términos h_ikm
h=zeros(777,1);
for i=1:7
    for k=1:7
        for m =1:7
            j=max([i k m]);
            for t=j:7
                h(100*i+10*k+m,1)=h(100*i+10*k+m,1)+
trace(U(:,:,10*t+10*k+m)*J(:,:,t)*(U(:,:,10*t+i).'));
            end
        end
    end
end

%% L8 Obtener la matriz columna de fuerzas centrípetas y de Coriolis H =
[Hi]'
H=zeros(7,1);
%     syms qp1 qp2 qp3 qp4 qp5 qp6 qp7

for i=1:7
    Aux =0;
    for k=1:7
        for m=1:7
            Aux= Aux+ h(100*i+10*k+m,1)*qpx(k,1)*qpx(m,1);
        end
    end
    H(i,1)= H(i,1)+ Aux;
end

%% L9 Obtener la matriz columna de fuerzas de gravedad C = [ci]

C=zeros(7,1);
for i=1:7
    for j=1:7
        C(i,1)= C(i,1)-(M(j,1)*g*U(:,:,10*j+i)*r(:,:,j));
    end
end

%% L10 Ecuación dinámica del sistema Tau = D*qpp + H + C
%     syms qpp1 qpp2 qpp3 qpp4 qpp5 qpp6 qpp7

Tau = D*qppx + H + C;
Tau = double(Tau);

```

9.2.6.2 Formulación Newton-Euler

```

%% MODELAMIENTO DINÁMICO INVERSO
% Algoritmo Newton - Euler
% -----

function Tau = MDI_NE(q, qp, qpp)
%% Condiciones Iniciales
% Velocidades angulares a partir de la posición y el Jacobiano

%% Paso 1: Asignar los sistemas DH

%% Paso 2: Condiciones Iniciales
t1=0; t2=q(1,1); t3=q(2,1); t4=q(3,1); t5=q(4,1); t6=q(5,1); t7=0;
[Vel_angular,T1,T2,T3,T4,T5,T6,T7,T1i,T2i,T3i,T4i,T5i,T6i,T7i] =
Jacobiano(t2,t3,t4,t5,t6);
qp1=0; qp2=qp(1,1)*pi/180; qp3=qp(2,1)*pi/180;
qp4=qp(3,1)*pi/180; qp5=qp(4,1)*pi/180; qp6=qp(5,1)*pi/180; qp7=0;
qpp1=0; qpp2=qpp(1,1)*pi/180; qpp3=qpp(2,1)*pi/180;
qpp4=qpp(3,1)*pi/180; qpp5=qpp(4,1)*pi/180; qpp6=qpp(5,1)*pi/180; qpp7=0;
fe=zeros(3,1); ne=zeros(3,1);

% Para la base {S0}
g=9.81; %[m/s^2]
w00 = [0,0,0]'; % velocidad angular
wp00 = [0,0,0]'; % aceleración angular
v00 = [0,0,0]'; % velocidad lineal
vp00 = -[0,0,-g]'; % aceleración lineal
z0 = [0,0,1]';

% pii=[a_i, d_i*sin(alpha_i),d*cos(alpha_i)]
% Vector que va desde {S_i-1} hasta {S_i} expresado en {S_i}
% p11 = [ 0; D1*sind(0); D1*cosd(0)];
% p22 = [ 0; L2*sind(-90); L2*cosd(-90)];
% p33 = [A3; 0; 0];
% p44 = [ 0; 0; 0];
% p55 = [ 0; L5*sind(90); L5*cosd(90)];
% p66 = [ 0; 0; 0];
% p77 = [ 0; L7*sind(0); L7*cosd(0)];
p11 = [ 0; 0; 0.95];
p22 = [ 0; -0.1; 0];
p33 = [0.5; 0; 0];
p44 = [ 0; 0; 0];
p55 = [ 0; 0.4; 0];
p66 = [ 0; 0; 0];
p77 = [ 0; 0; 0.45];

% sii=[SCx_i;SCy_i;SCz_i]
% Centro de masa del eslabón i respecto al sistema {S_i}
% Datos extraídos del las propiedades físicas de las piezas en
Solidworks
s11 = [ 0; 0; -0.1380];
s22 = [ 0; 0.0473; 0];
s33 = [-0.2994; 0; 0];
s44 = [ 0; 0; 0.0377];
s55 = [ 0; -0.1179; 0];
s66 = [ 0; 0; 0.0273];
s77 = [ 0; 0; -0.3116];

% mii= Masa del eslabón i [kg]

```

```

m11 = 5.9664;
m22 = 1.7791;
m33 = 3.5773;
m44 = 1.9849;
m55 = 1.3959;
m66 = 0.1200;
m77 = 0.7399;

% Iii=[Ixx Ixy Ixz; Iyx Iyy Iyz; Izx Izy Izz]
% Matriz de inercia del eslabón i expresado en un sistema paralelo a
% {S_i} y con el origen en el centro de masas del eslabón.
% Matriz de inercias L.
I11 = [ 0.1138      0      0 ;
        0  0.1138      0 ;
        0      0  0.0217 ];

I22 = [ 0.0042      0      0 ;
        0  0.0021      0 ;
        0      0  0.0046 ];

I33 = [ 0.0070      0      0 ;
        0  0.0946      0 ;
        0      0  0.0914 ];

I44 = [ 0.0047      0      0 ;
        0  0.0047      0 ;
        0      0  0.0016 ];

I55 = [ 0.0112      0      0 ;
        0  0.0006      0 ;
        0      0  0.0111 ];

I66 = [ 0.0001      0      0 ;
        0  0.0001      0 ;
        0      0  0.00001166 ];

I77 = [ 0.0031      0      0 ;
        0  0.0031      0 ;
        0      0  0.0001 ];

%% Paso 3: Obtener las matrices de rotación y sus inversas
%% Matrices de rotación
R11= T1(1:3,1:3);
R22= T2(1:3,1:3);
R33= T3(1:3,1:3);
R44= T4(1:3,1:3);
R55= T5(1:3,1:3);
R66= T6(1:3,1:3);
R77= T7(1:3,1:3);
R88=eye(3);
R11_inv= T1i(1:3,1:3);
R22_inv= T2i(1:3,1:3);
R33_inv= T3i(1:3,1:3);
R44_inv= T4i(1:3,1:3);
R55_inv= T5i(1:3,1:3);
R66_inv= T6i(1:3,1:3);
R77_inv= T7i(1:3,1:3);
R88_inv=eye(3);

```

```

%% Paso 4: Obtener la velocidad angular del sistema {S_i}
%   wii = Rii_inv*(wii_anterior+z0*tpi)
   w11 = R11_inv*w00; % Debido a que es de traslación
   w22 = R22_inv*(w11 + z0*qp2);
   w33 = R33_inv*(w22 + z0*qp3);
   w44 = R44_inv*(w33 + z0*qp4);
   w55 = R55_inv*(w44 + z0*qp5);
   w66 = R66_inv*(w55 + z0*qp6);
   w77 = R77_inv*(w66 + z0*qp7);

%% Paso 5: Obtener la aceleración angular del sistema {S_i}
%   wpii = Rii_inv*(wpii_anterior+z0*tppi) + cross(wii_anterior,z0*tpi);
   wp11 = R11_inv*wp00; % Debido a que es de traslación
   wp22 = R22_inv*(wp11 + z0*qpp2 + cross(w11,z0*qp2));
   wp33 = R33_inv*(wp22 + z0*qpp3 + cross(w22,z0*qp3));
   wp44 = R44_inv*(wp33 + z0*qpp4 + cross(w33,z0*qp4));
   wp55 = R55_inv*(wp44 + z0*qpp5 + cross(w44,z0*qp5));
   wp66 = R66_inv*(wp55 + z0*qpp6 + cross(w55,z0*qp6));
   wp77 = R77_inv*(wp66 + z0*qpp7 + cross(w66,z0*qp7));

%% Paso 6: Obtener la aceleración lineal del sistema {S_i}
%   vpii =
cross(wpii,pii)+cross(wii,cross(wii,pii))+Rii_inv*vpii_anterior
   vp11 =
R11_inv*(z0*qpp1+vp00)+cross(wp11,p11)+2*cross(w11,(R11_inv*z0*qp1))+cross
s(w11,cross(w11,p11)); % OJO W11 CAMBIO A WP11
   vp22 = cross(wp22,p22)+cross(w22,cross(w22,p22))+R22_inv*vp11;
   vp33 = cross(wp33,p33)+cross(w33,cross(w33,p33))+R33_inv*vp22;
   vp44 = cross(wp44,p44)+cross(w44,cross(w44,p44))+R44_inv*vp33;
   vp55 = cross(wp55,p55)+cross(w55,cross(w55,p55))+R55_inv*vp44;
   vp66 = cross(wp66,p66)+cross(w66,cross(w66,p66))+R66_inv*vp55;
   vp77 = cross(wp77,p77)+cross(w77,cross(w77,p77))+R77_inv*vp66;

%% Paso 7: Obtener la aceleración lineal del CG del eslabón i
%   aii = cross(wpii,sii)+cross(wii,cross(wii,sii))+vpii
   a11 = cross(wp11,s11)+cross(w11,cross(w11,s11))+vp11;
   a22 = cross(wp22,s22)+cross(w22,cross(w22,s22))+vp22;
   a33 = cross(wp33,s33)+cross(w33,cross(w33,s33))+vp33;
   a44 = cross(wp44,s44)+cross(w44,cross(w44,s44))+vp44;
   a55 = cross(wp55,s55)+cross(w55,cross(w55,s55))+vp55;
   a66 = cross(wp66,s66)+cross(w66,cross(w66,s66))+vp66;
   a77 = cross(wp77,s77)+cross(w77,cross(w77,s77))+vp77;

%% Paso 8: Obtener la fuerza ejercida sobre el eslabón i
%   fii = Rii_posterior*fii_posterior+mii*aii
   f77 = R88*fe + m77*a77;
   f66 = R77*f77 + m66*a66;
   f55 = R66*f66 + m55*a55;
   f44 = R55*f55 + m44*a44;
   f33 = R44*f44 + m33*a33;
   f22 = R33*f33 + m22*a22;
   f11 = R22*f22 + m11*a11;

%% Paso 9: Obtener el par ejercido sobre el eslabón i
%   nii = Rii_pos*[nii_pos+cross(Rii_pos_inv*prii,fii_pos)] + ...
%   cross(prii+srii,mii*aii) + Iii*wii+cross(wii,Iii*wii);
n77 = R88*(ne
+cross(R88_inv*p77,fe ))+cross((p77+s77),m77*a77)+I77*wp77+cross(w77,(I77
*w77));

```

```

n66 =
R77*(n77+cross(R77_inv*p66,f77))+cross((p66+s66),m66*a66)+I66*wp66+cross(
w66,(I66*w66));
n55 =
R66*(n66+cross(R66_inv*p55,f66))+cross((p55+s55),m55*a55)+I55*wp55+cross(
w55,(I55*w55));
n44 =
R55*(n55+cross(R55_inv*p44,f55))+cross((p44+s44),m44*a44)+I44*wp44+cross(
w44,(I44*w44));
n33 =
R44*(n44+cross(R44_inv*p33,f44))+cross((p33+s33),m33*a33)+I33*wp33+cross(
w33,(I33*w33));
n22 =
R33*(n33+cross(R33_inv*p22,f33))+cross((p22+s22),m22*a22)+I22*wp22+cross(
w22,(I22*w22));
n11 =
R22*(n22+cross(R22_inv*p11,f22))+cross((p11+s11),m11*a11)+I11*wp11+cross(
w11,(I11*w11));

%% Paso 10: Obtener la fuerza o par aplicado a la articulación i
% tau_i = nii'*(Rii_inv*z0);
tau1 = f11'*(R11_inv*z0);
tau2 = n22'*(R22_inv*z0);
tau3 = n33'*(R33_inv*z0);
tau4 = n44'*(R44_inv*z0);
tau5 = n55'*(R55_inv*z0);
tau6 = n66'*(R66_inv*z0);
tau7 = n77'*(R77_inv*z0);
Tau=[tau1;tau2;tau3;tau4;tau5;tau6;tau7];

```

9.2.6.3 Usando el Toolbox de Robótica de Peter Corke

```

%% MDI TOOLBOX DE ROBOTICA PETER CORKE- MÉTODO NEWTON EULER
% -----

clear all
clc

Esfera = Tl_Esferal(); % Define la trayectoria esferica
sim('Prueba_MG1.slx') % Obtiene el MGD
NQ = Q.Data(:, :); % posiciones angulares
NQp = FunQp(NQ); % velocidades angulares
NQpp = Redondear2(FunQp(NQp)); % aceleración angular
[r j]=size(NQ);

[NQt1,NQpt1,NQppt1] = Interpolar(NQ,NQp); % Se interpola y suaviza la
trayectoria
t=linspace(1,340,r); t=t'; % Cada tramo en 1 segundo
NQt.time=t;
NQt.signals.values = NQ;
NQt.signals.dimensions =6;
NQpt.time=t;
NQpt.signals.values = NQp;
NQpt.signals.dimensions =6;
NQppt.time=t;
NQppt.signals.values = NQpp;
NQppt.signals.dimensions =6;

sim('Prueba_MC1.slx')

```

```

%% Definimos las condiciones iniciales
D1=0.95; % base + eslabon 1(asumimos cte)
L2=0.1; % eslabon2
A3=0.5; % eslabon3
L5=0.4; % eslabon4 + eslabon5
L7=0.45; % eslabon6 + laparoscopia eslabon7
%DH = [THETA D A ALPHA sigma offset]
L(1)=Link([ 0 0 0 0 1 D1] , 'standard');
L(2)=Link([ 0 L2 0 -pi/2 0 0] , 'standard');
L(3)=Link([ 0 0 A3 0 0 -pi/2], 'standard');
L(4)=Link([ 0 0 0 pi/2 0 pi] , 'standard');
L(5)=Link([ 0 L5 0 pi/2 0 pi] , 'standard');
L(6)=Link([ 0 0 0 pi/2 0 -pi/2], 'standard');
L(7)=Link([ 0 L7 0 0 0 0] , 'standard');

L(1).m = 5.9664; L(2).m = 1.7791; L(3).m = 3.5773; % masa
L(4).m = 1.9849; L(5).m = 1.3959; L(6).m = 0.1200; L(7).m = 0.7399;

L(1).r = [ 0 0 -0.1380];
L(2).r = [ 0 0.0473 0];
L(3).r = [-0.2994 0 0]; % centro de gravedad
L(4).r = [ 0 0 0.0377];
L(5).r = [ 0 -0.1179 0];
L(6).r = [ 0 0 0.0273];
L(7).r = [ 0 0 -0.3116];

L(1).I = [ 0.1138 0.1138 0.0217];
L(2).I = [ 0.0042 0.0021 0.0046];
L(3).I = [ 0.0070 0.0946 0.0914]; %Tensor inercia (Ixx, Iyy, Izz)
L(4).I = [ 0.0047 0.0047 0.0016];
L(5).I = [ 0.0112 0.0006 0.0111];
L(6).I = [ 0.0001 0.0001 0.00001166];
L(7).I = [ 0.0031 0.0031 0.0001];

L(1).Jm = 0; L(2).Jm = 0; L(3).Jm = 0; % Inercia del motor actuador
L(4).Jm = 0; L(5).Jm = 0; L(6).Jm = 0; L(7).Jm = 0;

L(1).G = 1; L(2).G = 1; L(3).G = 1; % Coef. de reducci3n actuador
L(4).G = 1; L(5).G = 1; L(6).G = 1; L(7).G = 1;

L(1).B = 0; L(2).B = 0; L(3).B = 0; % Rozamiento viscoso
L(4).B = 0; L(5).B = 0; L(6).B = 0; L(7).B = 0;

L(1).Tc = [0 0]; L(2).Tc = [0 0]; L(3).Tc = [0 0]; % Roz.de Coulomb
L(4).Tc = [0 0]; L(5).Tc = [0 0]; L(6).Tc = [0 0]; L(7).Tc = [0 0];

IMEbot=SerialLink(L, 'name', 'IMEbot');

%% Graficar la trayectoria del robot
W = [-0.3, 0.8, -0.3, 0.3, -0.1, 1.7];
for i=1:1:r
    p2=(NQ(i,1))*pi/180;
    p3=(NQ(i,2))*pi/180;
    p4=(NQ(i,3))*pi/180;
    p5=(NQ(i,4))*pi/180;
    p6=(NQ(i,5))*pi/180;
    hold on
    [Pos(i,1),Pos(i,2), Pos(i,3)] =
MGD(NQ(i,1),NQ(i,2),NQ(i,3),NQ(i,4),NQ(i,5),NQ(i,6));
    plot3(Pos(:,1),Pos(:,2), Pos(:,3), 'b');

```

```

%           axis([-0.3, 0.8, -0.3, 0.3, -0.1, 1.7])
%           IMEbot.plot([0 p2 p3 p4 p5 p6 0], 'workspace', W)
%           hold off;
%           grid on
%           pause(0.01)
%           end

%% Calculando los torques
grav = [0,0,9.81];
NewQ  = [zeros(r,1) NQ*pi/180];
NewQp = [zeros(r,1) NQp*pi/180];
NewQpp = [zeros(r,1) NQpp*pi/180];
Fext=[0 0 0 0 0 0];
Tau  = IMEbot.rne(NewQ,NewQp,NewQpp,grav,Fext);%
TauD = IMEbot.itorque(NewQ,NewQpp);
TauH1 = IMEbot.coriolis(NewQ,NewQp);
for i=1:r
TauH(i,:) = TauH1(:, :, i) * (NewQp(i, :)' );
end
TauC = IMEbot.gravload(NewQ);
Taucomp=TauD+TauH+TauC;
TauI1 = IMEbot.inertia(NewQ);
for i=1:r
TauI(i,:) = TauI1(:, :, i) * (NewQpp(i, :)' );
qpp_1(i,:) = ((TauI1(:, :, i) ^ (-1)) * [Tau(i, :)' -TauC(i, :)' -
TauH(i, :)' ]) * 180/pi;
end
T=zeros(340,7);
T(1,:)=Tau(1,:);
for i=2:340
    T(i,:)=Tau(i,:)-Tau(i-1,:);
end
Taut.time=t;
Taut.signals.values = Tau;
Taut.signals.dimensions =7;

IME.time=t;
IME.signals.values = T;
IME.signals.dimensions =7

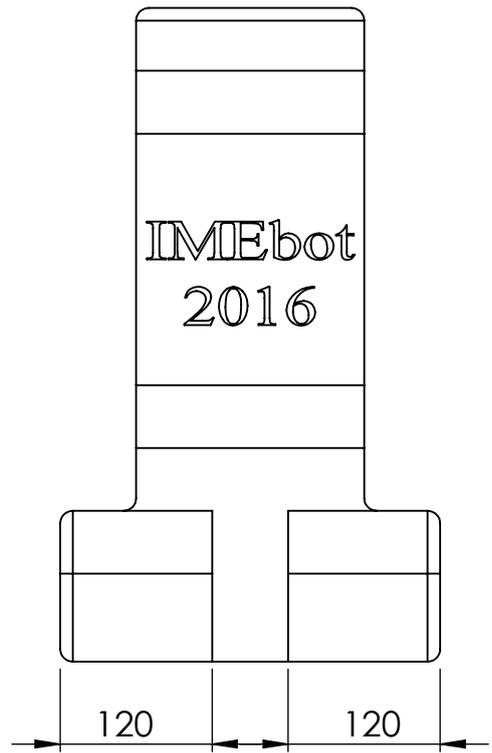
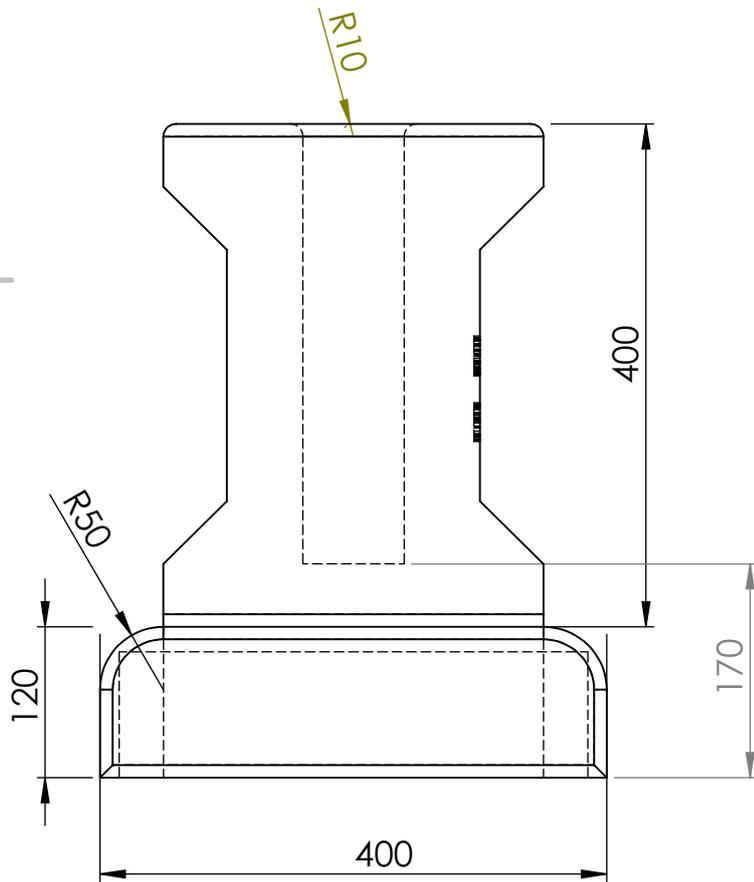
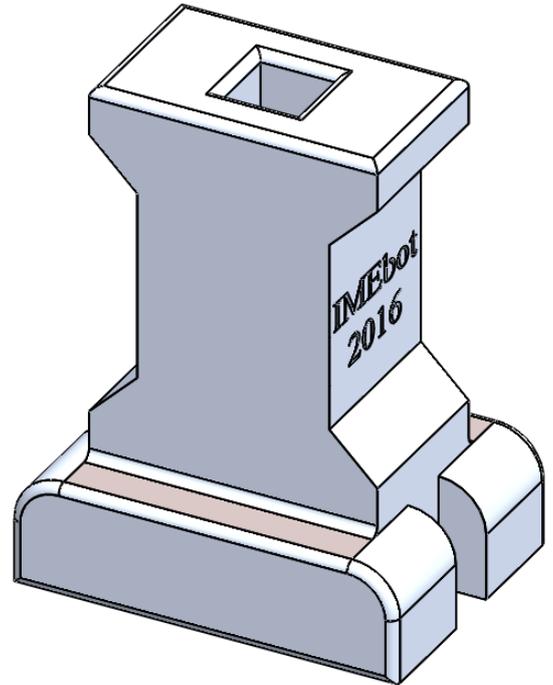
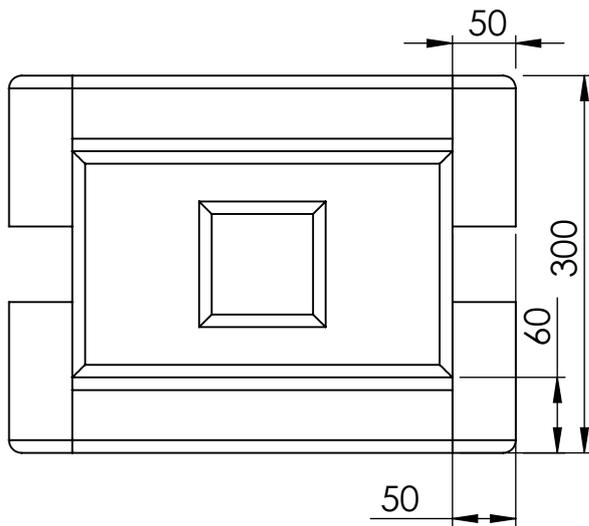
sim('Prueba_MD1.slx')

```


Planos

En este apartado se presentan los planos a detalle de los eslabones que constituyen el asistente robótico IMEbot:

- PLANO 01/08: Base
- PLANO 02/08: Eslabón 1
- PLANO 03/08: Eslabón 2
- PLANO 04/08: Eslabón 3
- PLANO 05/08: Eslabón 4
- PLANO 06/08: Eslabón 5
- PLANO 07/08: Eslabón 6
- PLANO 08/08: Eslabón 7 - Laparoscopio



| | FECHA | NOMBRE |
|----------|---------|----------|
| DIBUJADO | DIC. 16 | M. AMAYA |
| REVISADO | FEB 17 | C. OJEDA |

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO:

Base

ESCALA:

S/E

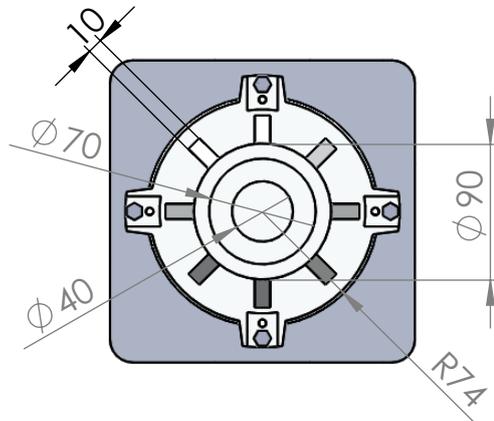
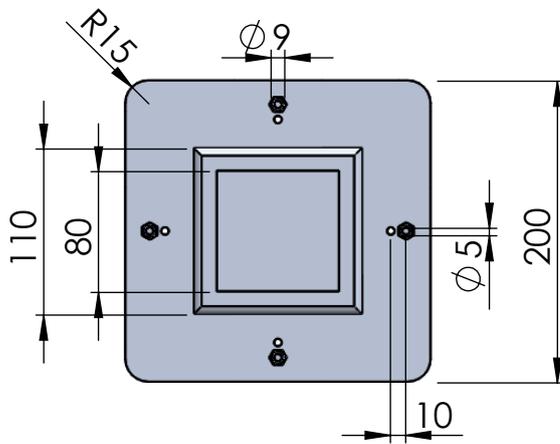
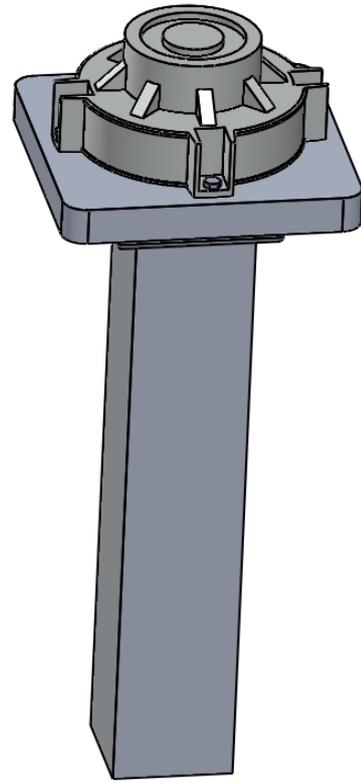
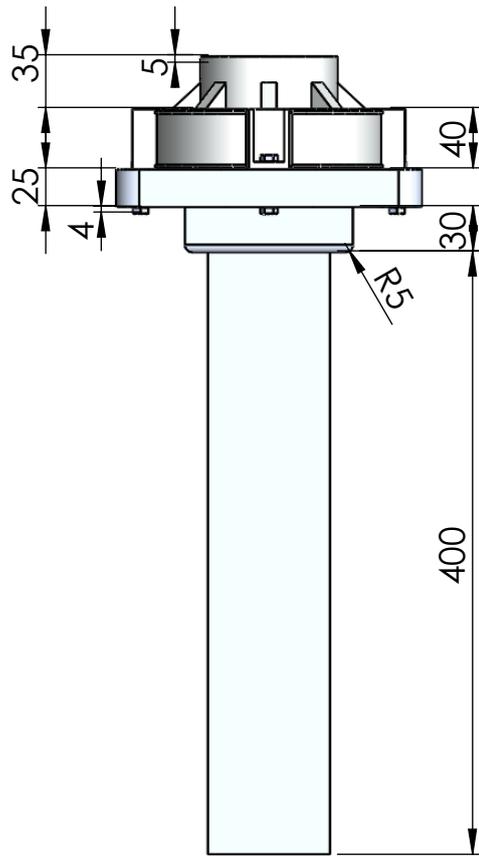
NOMBRES Y APELLIDOS:

Manuel Amaya González

IMEbot

PLANO N°:

01 de 08



| | FECHA | NOMBRE |
|----------|---------|----------|
| DIBUJADO | DIC. 16 | M. AMAYA |
| REVISADO | FEB 17 | C. OJEDA |

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO:

Eslabón 1

ESCALA:

S/E

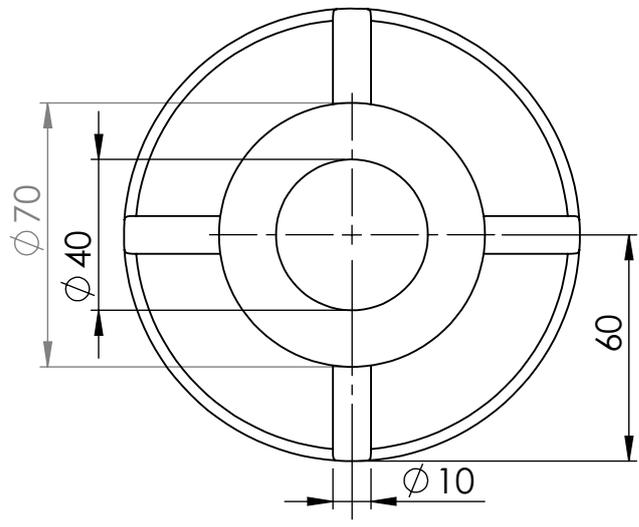
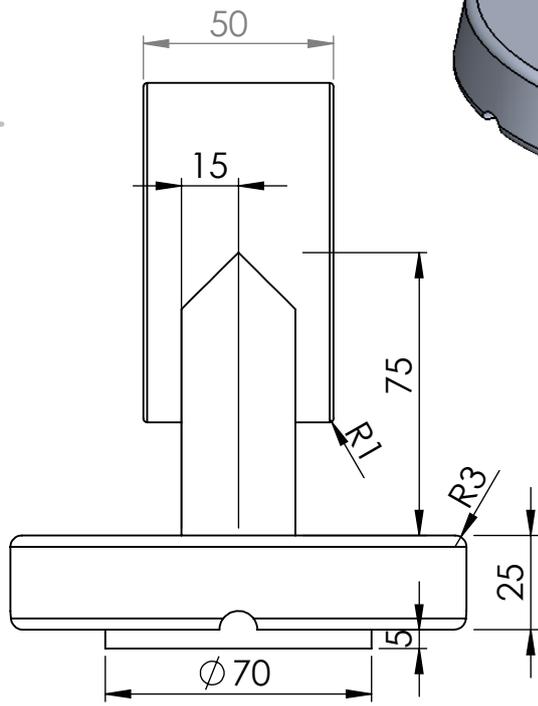
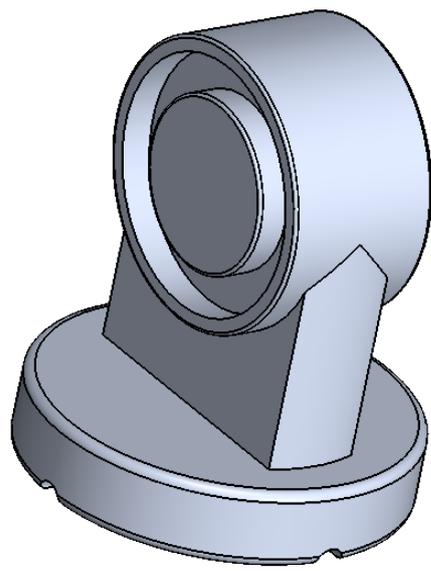
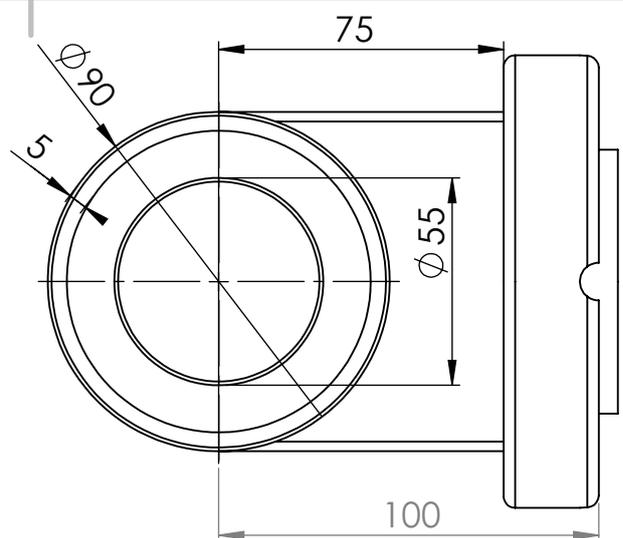
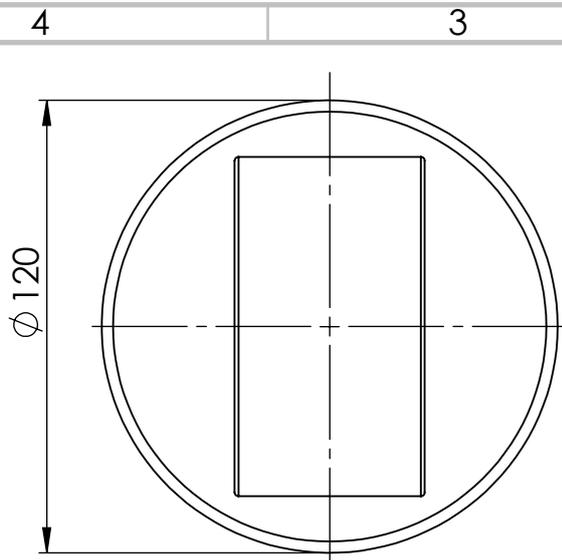
NOMBRES Y APELLIDOS:

Manuel Amaya González

IMEbot

PLANO N°:

02 de 08



| | FECHA | NOMBRE |
|----------|---------|----------|
| DIBUJADO | DIC. 16 | M. AMAYA |
| REVISADO | FEB 17 | C. OJEDA |

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

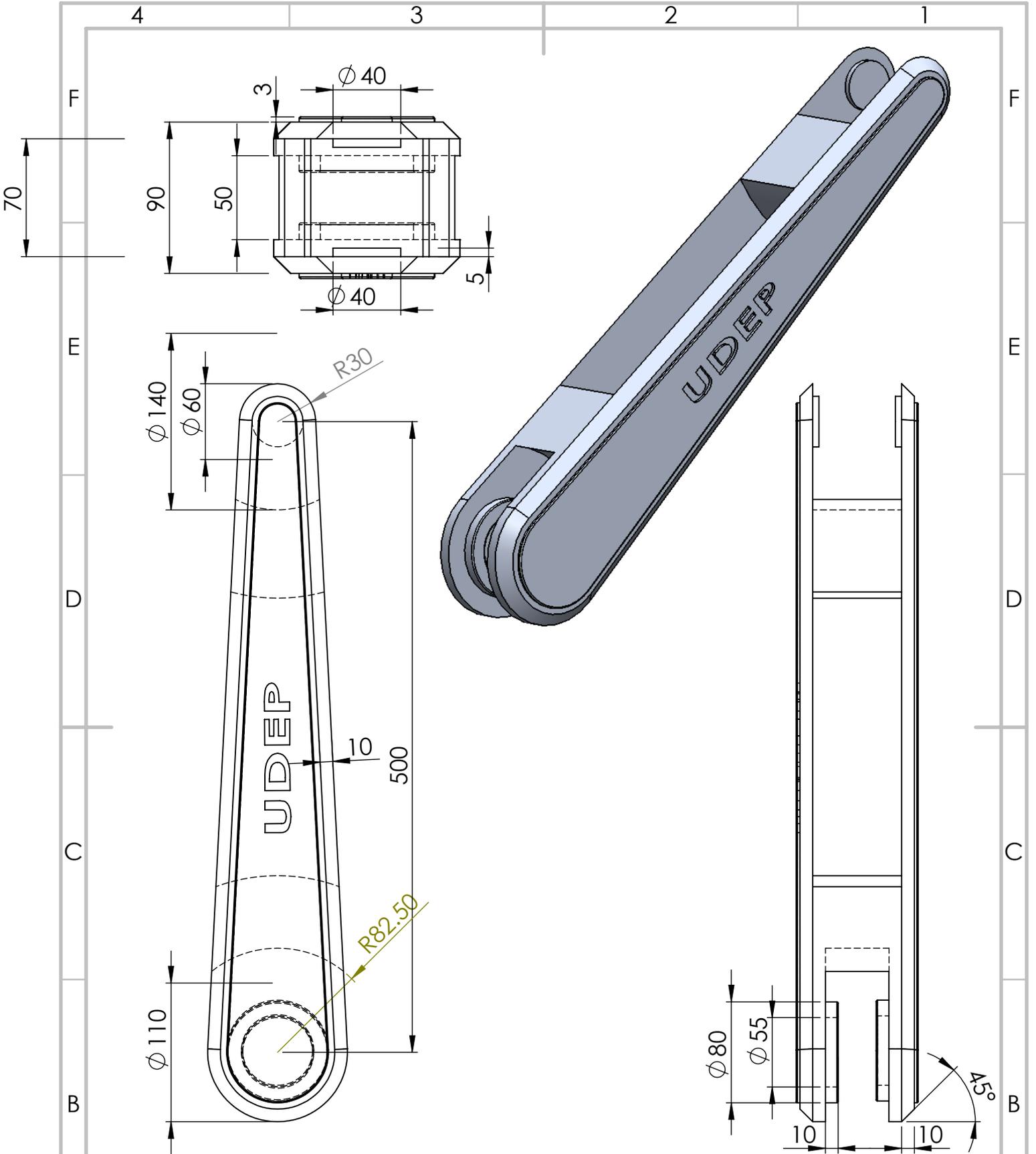
TÍTULO:
Eslabón 2

ESCALA:
S/E

NOMBRES Y APELLIDOS:
Manuel Amaya González

IMEbot

PLANO N°:
03 de 08



| | FECHA | NOMBRE |
|----------|---------|----------|
| DIBUJADO | DIC. 16 | M. AMAYA |
| REVISADO | FEB 17 | C. OJEDA |

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO:
Eslabón 3

NOMBRES Y APELLIDOS:
Manuel Amaya González

IMEbot

ESCALA:
S/E

PLANO N°:
04 de 08

4

3

2

1

F

F

E

E

D

D

C

C

B

B

A

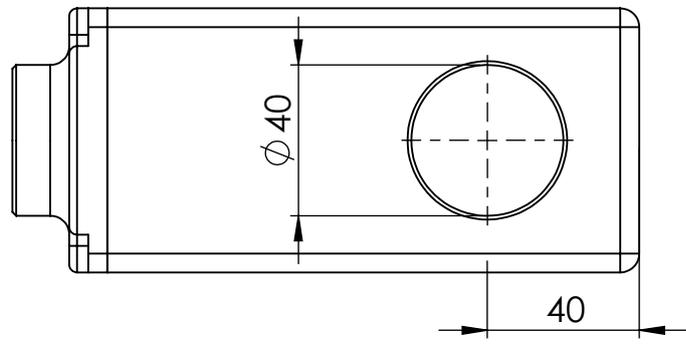
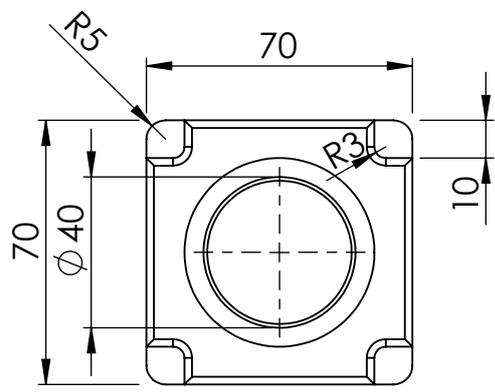
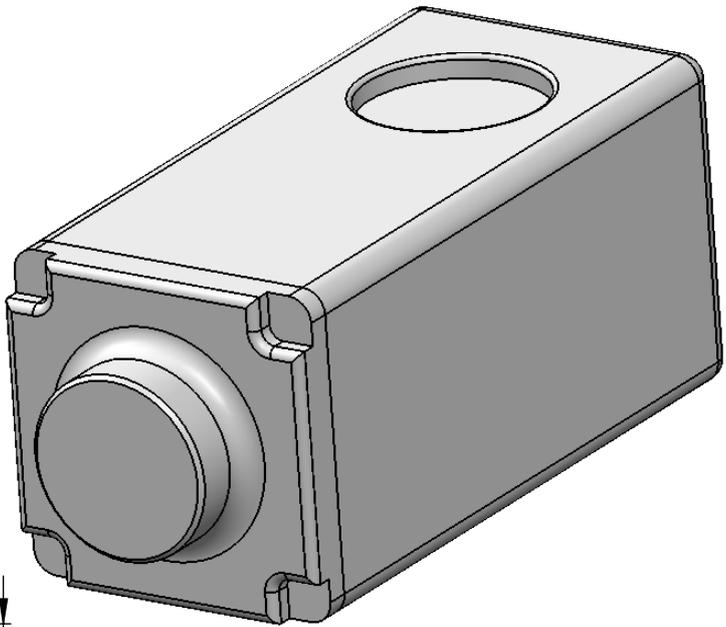
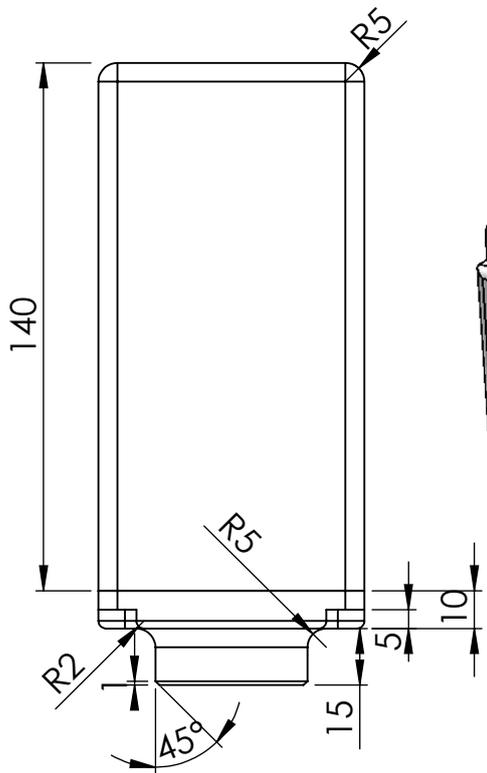
A

4

3

2

1



| | FECHA | NOMBRE |
|----------|---------|----------|
| DIBUJADO | DIC. 16 | M. AMAYA |
| REVISADO | FEB 17 | C. OJEDA |

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

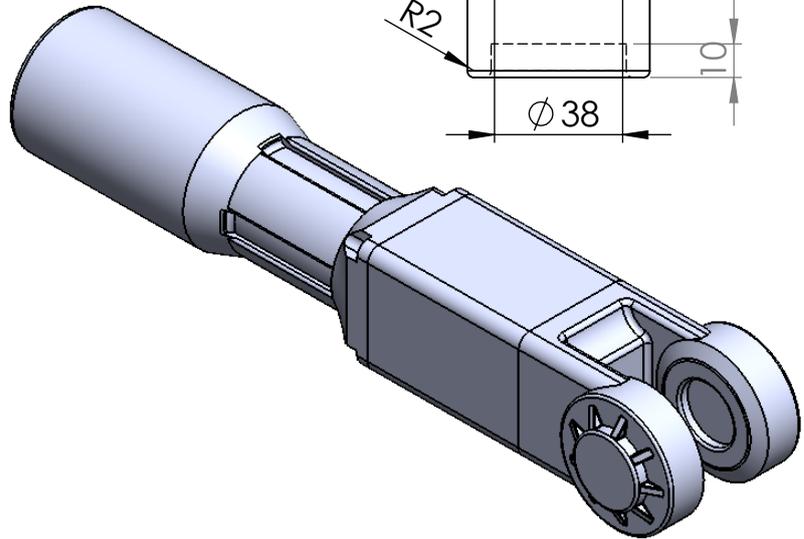
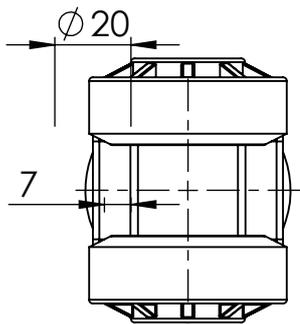
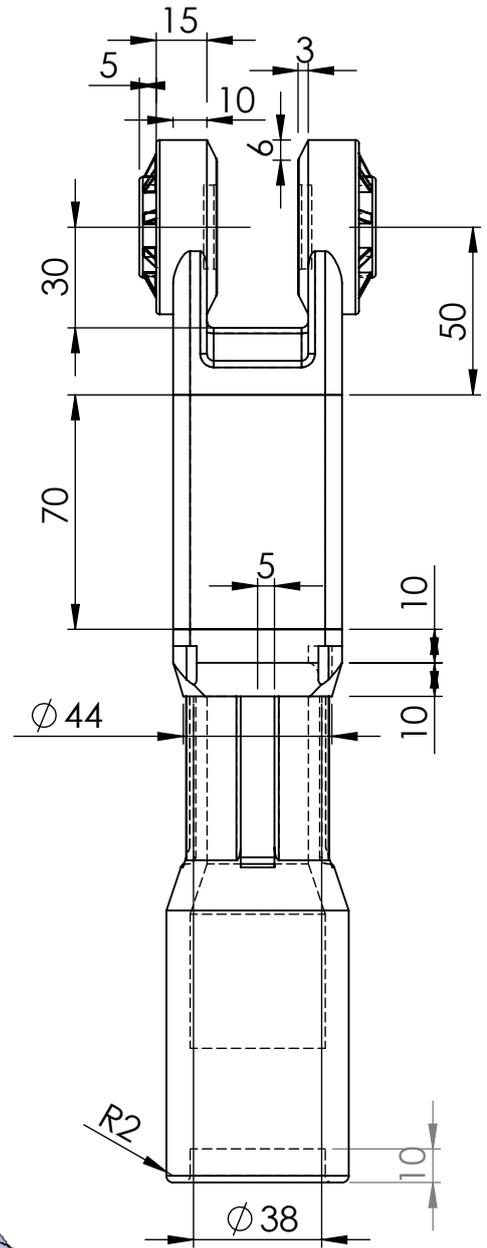
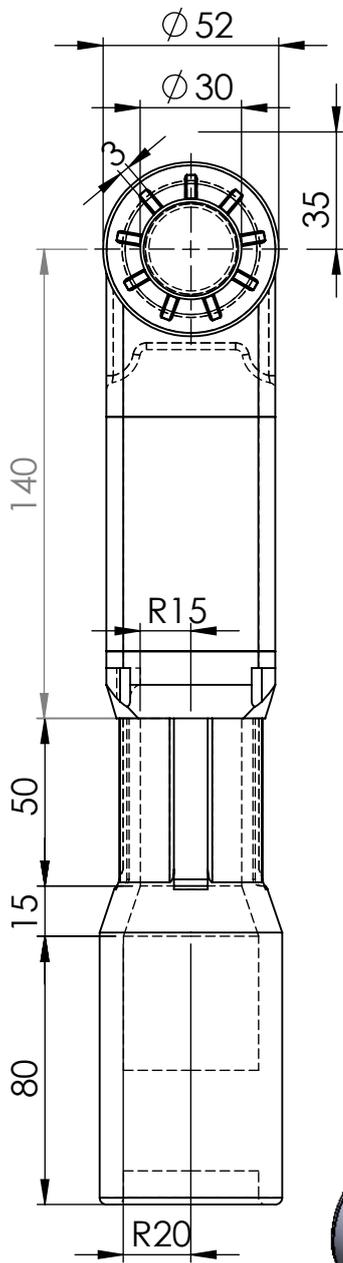
TÍTULO:
Eslabón 4

ESCALA:
S/E

NOMBRES Y APELLIDOS:
Manuel Amaya González

IMEbot

PLANO N°:
05 de 08



| | FECHA | NOMBRE |
|----------|---------|----------|
| DIBUJADO | DIC. 16 | M. AMAYA |
| REVISADO | FEB 17 | C. OJEDA |

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO:

Eslabón 5

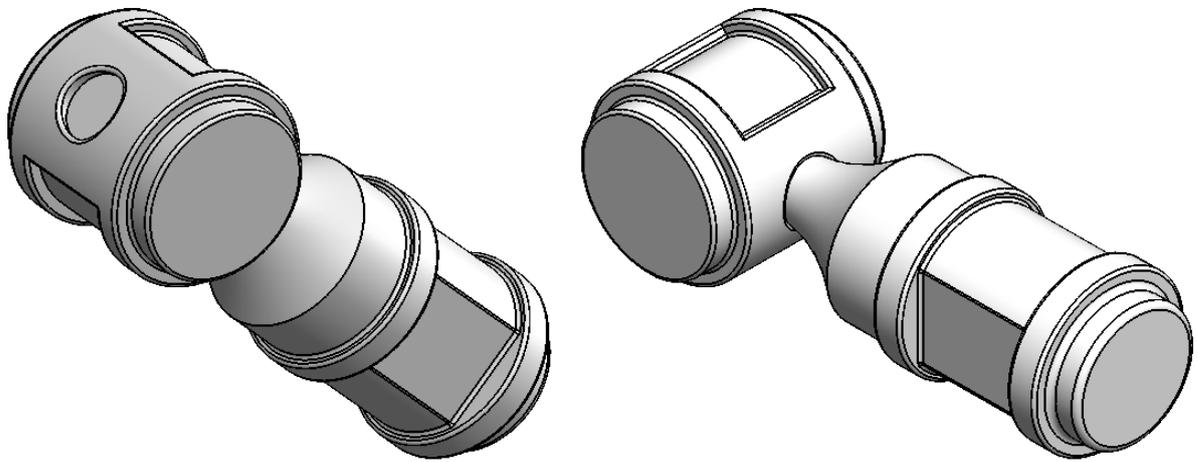
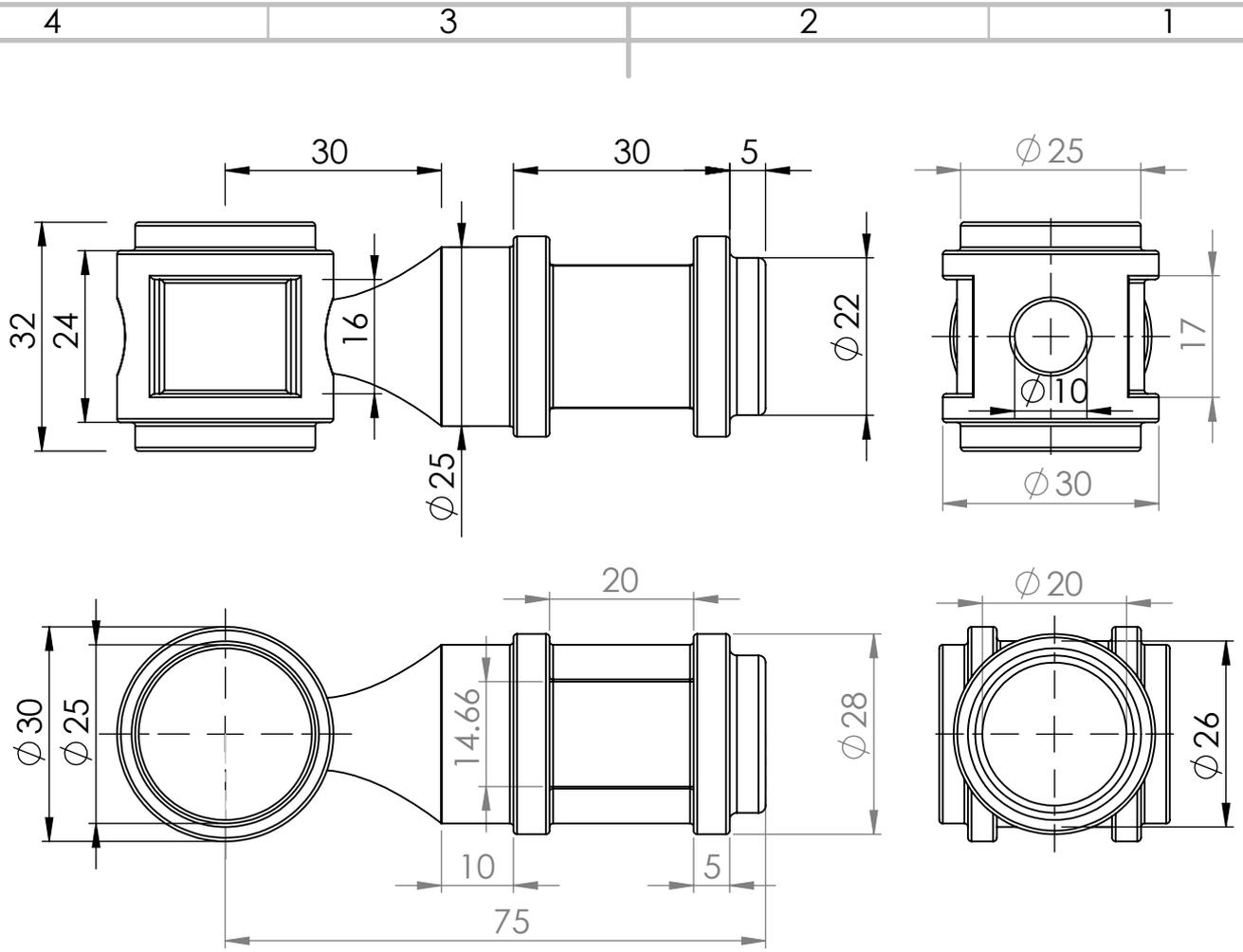
ESCALA:
S/E

NOMBRES Y APELLIDOS:

Manuel Amaya González

IMEbot

PLANO N°:
06 de 08



| | FECHA | NOMBRE |
|----------|---------|----------|
| DIBUJADO | DIC. 16 | M. AMAYA |
| REVISADO | FEB 17 | C. OJEDA |

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO:

Eslabón 6

ESCALA:

S/E

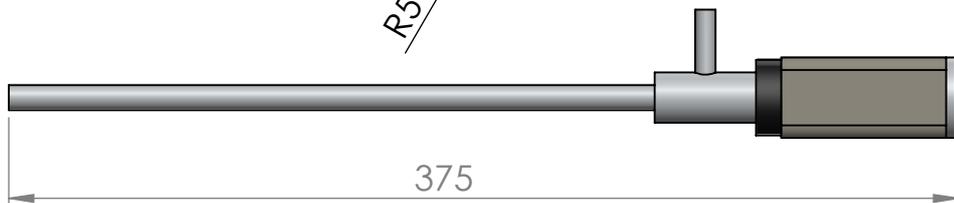
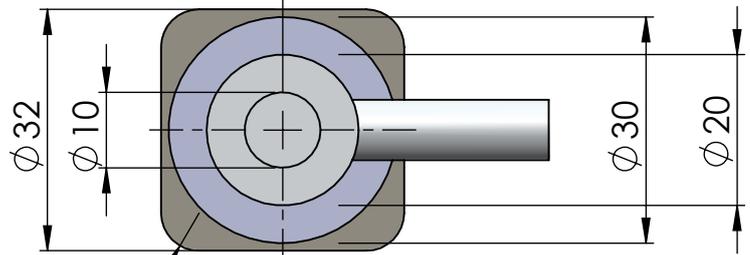
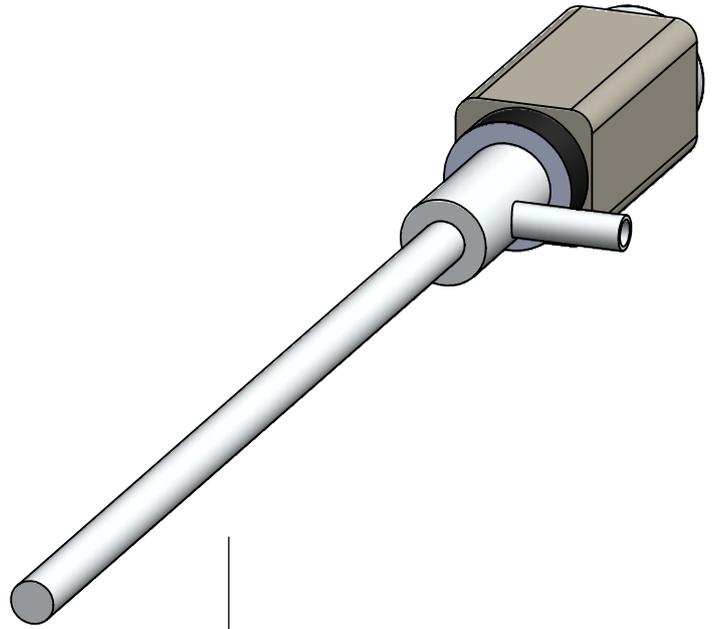
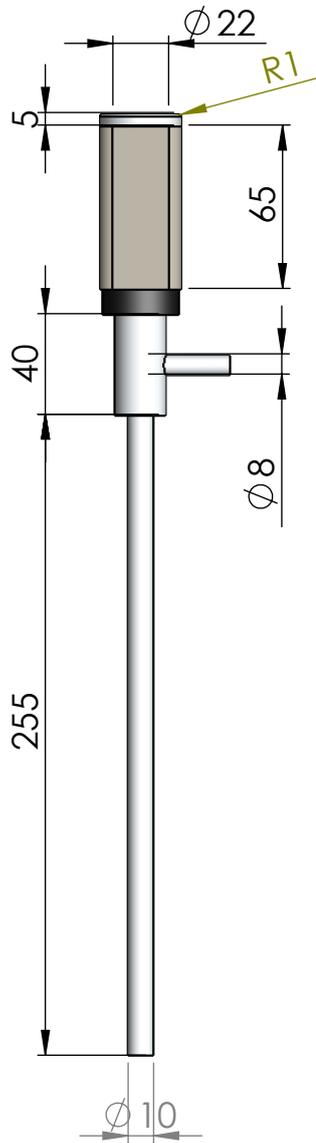
NOMBRES Y APELLIDOS:

Manuel Amaya González

IMEbot

PLANO N°:

07 de 08



| | FECHA | NOMBRE |
|----------|---------|----------|
| DIBUJADO | DIC. 16 | M. AMAYA |
| REVISADO | FEB 17 | C. OJEDA |

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO:

Eslabón 7
Laparoscopio

ESCALA:
S/E

NOMBRES Y APELLIDOS:

Manuel Amaya González

IMEbot

PLANO N°:
08 de 08