



UNIVERSIDAD
DE PIURA

REPOSITORIO INSTITUCIONAL
PIRHUA

DISEÑO Y ENSAMBLE DE UN BRAZO ROBOT COMO MÓDULO DE LABORATORIO PARA EL ESCANEEO DE CURVAS EN 3D

Marco Troncos-Riofrio

Piura, diciembre de 2016

Facultad de Ingeniería

Departamento de Ingeniería Mecánico-Eléctrica

Troncos, M. (2016). *Diseño y ensamble de un brazo robot como módulo de laboratorio para el escaneo de curvas en 3D* (Tesis de pregrado en Ingeniería Mecánico-Eléctrica). Universidad de Piura, Facultad de Ingeniería. Programa Académico de Ingeniería Mecánico-Eléctrica. Piura, Perú.



Esta obra está bajo una [licencia](#)
[Creative Commons Atribución-](#)
[NoComercial-SinDerivadas 2.5 Perú](#)

[Repositorio institucional PIRHUA – Universidad de Piura](#)

UNIVERSIDAD DE PIURA

FACULTAD DE INGENIERÍA



“Diseño y ensamble de un brazo robot como módulo de laboratorio para el escaneo de curvas en 3D”

Tesis para optar el Título de
Ingeniero Mecánico - Eléctrico

Marco Aurelio Troncos Riofrio

Asesor: Dr. Ing. Miguel Buenaventura Castro Sánchez.

Piura, Diciembre 2016

Dedicatoria

Este trabajo está dedicado a Dios, de quién he aprendido que debo vivir con humildad todos los proyectos de mi vida; a mis padres Aurelio y Carmen, por el amor brindado en todo momento; a mi hermana Aissa, por su cariño y sabias enseñanzas; y a mis compañeros y amigos, que me apoyaron durante el desarrollo de este trabajo.

Prólogo

Aunque existen varias definiciones de “robot industrial” todas coinciden en que son dispositivos mecánicos con capacidad de manipulación y, por lo tanto, están dotados de uno o más brazos mecánicos independientes que pueden ser controlados por un ordenador. Por su configuración un robot industrial es un manipulador formado por tres sistemas básicos: la estructura mecánica que consiste en eslabones, juntas y actuadores al final del brazo mecánico; el sistema de control que puede ser fijo o servo; y las unidades de potencia, que pueden ser hidráulicas, neumáticas o eléctricas.

Al hablar de un manipulador se nos viene a la mente un brazo cuyo actuador es la mano, los eslabones son el mismo brazo y el antebrazo, y las juntas son las articulaciones; todo ello operado por un sistema de control llamado cerebro. Así, el brazo mecánico permite reemplazar la mano, con la ventaja de operar a mayor velocidad y gran precisión; realizar tareas riesgosas para el ser humano como es la manipulación de sustancias radiactivas.

Por otro lado, un brazo posee tecnología transdisciplinaria, porque hace uso de varias ciencias afines como son: matemáticas, física, electrónica, computación, sistemas automáticos, etc., es por ello que posee un gran potencial en la didáctica de las enseñanzas de las ciencias básicas de la Facultad de Ingeniería y de los cursos de carrera en los alumnos de Ingeniería Mecánico – Eléctrica. Un prototipo de brazo robótico permitirá visualizar cómo un simple manipulador mecánico se puede convertir en un robot con control sensorizado, que capta información del entorno para ser procesada con un sencillo lenguaje de programación y con un entorno amigable.

El presente trabajo de tesis denominado “Diseño y ensamble de un brazo robot como módulo de laboratorio para el escaneo de curvas en 3D” constituye una aplicación práctica de los conceptos físicos y matemáticos estudiados durante la carrera. El uso de transductores rotativos permite transformar el movimiento angular en una serie de pulsos digitales que son leídos en un software libre llamado Arduino y procesados en Matlab, un entorno de cálculo numérico y de visualización, que muestra la representación de la curva palpada por el brazo robótico, haciendo las veces de un escáner; todo ello con el objetivo de introducir a los alumnos de ingeniería al conocimiento y aplicación de controladores numéricos.

Este trabajo no hubiera sido posible sin el apoyo recibido en consejos, comentarios, sugerencias, elogios y críticas constructivas de los ingenieros Miguel Castro, Jorge Yaksetig Castillo y Víctor Lizana. La construcción del prototipo merece un reconocimiento especial para el personal técnico del Laboratorio de Mecánica de la Universidad de Piura que con su granito de arena contribuyeron a la realización de este proyecto.

Resumen

El presente trabajo de investigación tiene como objetivo diseñar y construir un brazo manipulador acondicionado para la obtención de curvas en el espacio virtual a través de una interfaz gráfica en Matlab, controlado automáticamente con Arduino. Para lograr este objetivo se construyó un brazo mecánico con cinco grados de libertad al que se le acoplaron cinco encoders ópticos incrementales que generan señales digitales de posición, y cuya información se procesa en Arduino, una sencilla plataforma de hardware para la recepción de datos, cuya representación gráfica se logra a través de Matlab, un entorno de cálculo numérico y de visualización. El resultado de este trabajo es un módulo de laboratorio que escanea curvas, pero con ciertas limitaciones de precisión. Los errores son del orden del 3% para curvas sencillas y 11% para curvas complicadas; sin embargo, se consolida como un prototipo de laboratorio que introduce al alumno de pregrado al conocimiento de la robótica mediante la adaptación de un sencillo brazo mecánico a un manipulador adaptado electrónicamente para cubrir una necesidad como es el escaneo de curvas, haciendo uso de modernos lenguajes de programación y, sobre todo, aprovechando equipos y recursos existentes en los laboratorios de mecánica y electrónica.

Índice General

Introducción	1
Capítulo 1. Marco teórico.....	3
1.1 Antecedentes históricos.....	3
1.2 Definición y clasificación de los robots	4
1.2.1 Definición del robot industrial	4
1.2.2 Clasificación del robot industrial	5
1.2.3 Clasificación según el tipo de trayectoria.....	6
1.2.4 Clasificación según AFRI	6
1.2.5 Clasificación según IFR	7
1.2.6 Clasificación según Knasel	7
1.3 Morfología de los Manipuladores	8
1.3.1 Grados de libertad	8
1.3.2 Estructuras básicas	8
1.3.3 Articulaciones.....	9
1.3.4 Acondicionamiento de un brazo manipulador para lectura de posición.....	10
1.4 Sistema de adquisición de datos.....	10
1.4.1 Transductores	11
1.4.2. Encoders.....	15
Capítulo 2. Fundamentos de Robótica	17
2.1 Herramientas matemáticas para la localización espacial.....	17
2.1.1 Representación de la posición	17
2.1.2 Representación de la orientación.....	19
2.2 Comparación de métodos de localización espacial	33
2.3 Cinemática de un robot	33
2.3.1 Problema cinemático directo.....	34
2.3.2 Cinemática inversa	42
Capítulo 3. Diseño y ensamble del brazo robot.....	43
3.1 Confrontación del problema.....	43
3.1.1 Formulación del problema	43
3.1.2 Análisis FODA.....	43
3.2 Selección de los conceptos de diseño y construcción	45

3.2.1 Fundamento de los cinco grados de libertad	45
3.2.2 Análisis estático del brazo robot	46
3.2.3 Selección de los componentes principales	50
3.2.4 Componentes secundarios	55
3.2.5 Trabajos de mecanizado y ensamble del brazo robot.....	55
3.2.6. Listado y costo de materiales	59
3.2.7. Especificaciones de trabajo	60
Capítulo 4. Programación para el control del brazo robot.....	63
4.1 Arduino	63
4.1.1 Arduino Nano - Hardware.....	63
4.1.2 Software	64
4.1.3 Diagrama de flujo de la programación en Arduino	64
4.2 Matlab	66
4.2.1 Código base de programación	66
4.2.2 Diagrama de flujo de la programación de la GUI.	67
4.2.3 Descripción de la GUI.....	68
Capítulo 5. Manual de uso del brazo robot y realización de ensayos.....	77
5.1 Guía de usuario	77
5.1.1 Descripción de los botones de la GUI.....	77
5.1.2 Establecer conexión robot – PC	78
5.1.3 Colocación de la posición inicial.....	79
5.1.4 Captura de datos	79
5.1.5 Ajuste de curva en Solidworks.....	81
5.2 Realización de ensayos.....	82
5.2.1 Ensayo de escaneo de curva de prueba	82
5.2.2 Ensayo de escaneo de una circunferencia	84
5.2.3 Ensayo de escaneo del perímetro de una pieza rectangular	86
Conclusiones	89
Recomendaciones.....	91
Bibliografía	93
Anexos.....	935

Índice de tablas

Tabla 1.1. Clasificación de los robots según la AFRI	6
Tabla 1.2. Clasificación de los robots industriales por generaciones	7
Tabla 1.3. Clasificación de los robots según Knasel	7
Tabla 1.4. Transductores de diversas magnitudes físicas.	13
Tabla 2.1. Valores de ingreso para prueba 1.	40
Tabla 3.1. Análisis FODA aplicado al prototipo de brazo robótico	45
Tabla 3.2. Cálculo del centro de gravedad del prototipo de brazo robot.	47
Tabla 4.1. Características del protoboard.	63

Índice de figuras

Figura	1.1	Configuraciones básicas de robots manipuladores industriales	8
Figura	1.2	Representación gráfica de las señales incrementales A, B y Z	10
Figura	1.3	Forma general de un sistema de medida	11
Figura	1.4	Estructura genérica de transductor	11
Figura	1.5	Encoder incremental	15
Figura	1.6	Discriminador de posición y sentido de giro.	16
Figura	2.1	Representación de un vector en coordenadas cartesianas	18
Figura	2.2	Representación de un vector en coordenadas polares y cilíndricas	18
Figura	2.3	Representación de un vector en coordenadas esféricas	19
Figura	2.4	Orientación de un sistema OUV respecto a otro OXY en un plano	20
Figura	2.5	Sistema de referencia OXYZ y solidario objeto OUVW	21
Figura	2.6	Rotación del sistema OUVW con respecto a los ejes OY y OZ	22
Figura	2.7	Sistemas finales según el orden de las transformaciones	27
Figura	2.8	Gráfico de transformación	32
Figura	2.9	Diagrama de relación entre cinemática directa e inversa	34
Figura	2.10	Parámetros D-H para un eslabón giratorio.	37
Figura	2.11	Formulación Denavit – Hartenberg. Posición inicial del brazo robot	38
Figura	2.12	Identificación de las articulaciones y selección del sistema de referencia X_0, Y_0, Z_0 .	38
Figura	2.13	Generación de una articulación ficticia X_1, Y_1, Z_1	39
Figura	2.14	Sistema de referencia para el efector final	40
Figura	3.1	Factores clave del FODA	44
Figura	3.2	Similitud del brazo robot con la anatomía humana	46
Figura	3.3	Prototipo de brazo robótico para escaneo de curvas	46
Figura	3.4	Posiciones de análisis estático del brazo robot	47
Figura	3.5	Diagrama de cuerpo libre de la base	49
Figura	3.6	Encoder incremental - GHS38 – 1200 pulsos/revolución	50
Figura	3.7	Placa electrónica basada en Arduino Nano	51
Figura	3.8	Piezas construidas con ertalyte	52
Figura	3.9	Montaje de perno de sujeción	52
Figura	3.10	Instalación del encoder en el perno	53
Figura	3.11	Selección de un perno comercial	53
Figura	3.12	Montaje de rodamientos	54
Figura	3.13	Selección de rodamiento	54
Figura	3.14	Ensamble del disco y tapa base dentro del cilindro base	56
Figura	3.15	Ensamble de rodajes y bocina en tapa base	56
Figura	3.16	Ensamblaje del eje del encoder en el tornillo M12	57
Figura	3.17	Explosionado de piezas en una articulación del brazo robot	57
Figura	3.18	Ensamble de las dos primeras articulaciones del brazo robot	58
Figura	3.19	Vista de sección de las dos primeras articulaciones	58
Figura	3.20	Desmontaje de la placa de unión encoder – pieza	59
Figura	3.21	Ensamble final del brazo robot	59
Figura	3.22	Posición de trabajo 1	60
Figura	3.23	Variantes de la posición de trabajo 1	60
Figura	3.24	Posición de trabajo 2	61

Figura 3.25	Variantes de la posición de trabajo 2	61
Figura 4.1	Diagrama de flujo para código Arduino	65
Figura 4.2	Código inicial desarrollado en Matlab	67
Figura 4.3	Diagrama de flujo código Matlab	68
Figura 4.4	Comandos importantes para el diseño de una GUI	69
Figura 4.5	Sección de objetos dentro de la GUI	70
Figura 4.6	Pasos para visualizar el código de programación del objeto	70
Figura 4.7	Diseño de la GUI – Objetos utilizados con su respectivo Tag	71
Figura 4.8	Modificación 1 de la característica de un objeto	72
Figura 4.9	Modificación 2 de la característica de un objeto	72
Figura 4.10	Líneas de código de inicialización de la GUI	72
Figura 4.11	Condiciones iniciales generadas por Matlab	73
Figura 4.12	Condiciones iniciales de la GUI del brazo robot	73
Figura 4.13	Código para establecer comunicación serial Arduino – Matlab	74
Figura 4.14	Código para establecer comunicación serial Arduino – Matlab	75
Figura 4.15	Mensaje para una comunicación serial exitosa	75
Figura 4.16	Código para extraer los ángulos de la trama de datos de Arduino	76
Figura 5.1	Interfaz gráfica para el procesamiento de datos en Matlab	77
Figura 5.2	Posición inicial de referencia	79
Figura 5.3	Interfaz gráfica en funcionamiento	80
Figura 5.4	Exportar datos de Matlab en formato txt	80
Figura 5.5	Ruta para ajustar curva a una data de puntos	81
Figura 5.6	Pasos para ajustar curva a una data de puntos	81
Figura 5.7	Primera prueba a desarrollar	82
Figura 5.8	Ajuste de una curva por la data de puntos capturados	82
Figura 5.9	Comparación entre la curva de ajuste y la curva teórica de la prueba 1	83
Figura 5.10	Área del Ensayo - curva de prueba	83
Figura 5.11	Escaneo de la circunferencia externa de un disco	84
Figura 5.12	Ajuste de una curva por los puntos capturados en la prueba 2	84
Figura 5.13	Comparación de la curva teórica con la curva de ajuste de la prueba 2	85
Figura 5.14	Área del ensayo – escaneo de una circunferencia	85
Figura 5.15	Escaneo del perímetro de una pieza rectangular	86
Figura 5.16	Ajuste de una curva por los puntos capturados en la prueba 3	86
Figura 5.17	Comparación de la curva teórica con la curva de ajuste de la prueba 3	87
Figura 5.18	Área del ensayo – escaneo del perímetro de una pieza rectangular	87

Introducción

El presente trabajo de investigación tiene como objetivo diseñar y construir un brazo manipulador acondicionado electrónicamente para la obtención de curvas en el espacio virtual a través de una interfaz gráfica en Matlab, controlado con Arduino, y ha sido diseñado en cinco capítulos:

El primer capítulo, referido al marco teórico, presenta los antecedentes históricos de la robótica industrial; luego se describe la morfología de los robots manipuladores: grados de libertad, estructuras básicas y articulaciones; finalmente se introduce la teoría de los sistemas de adquisición de datos, poniendo énfasis en los sensores de lectura y los encoders.

En el segundo capítulo, fundamentos de robótica, se analiza la matriz de cambio de orientación para una comprensión básica de los principios de un brazo manipulador; la composición de rotaciones como base para comprender el principio matemático del brazo robótico manipulador; la matriz de transformación homogénea que permite establecer una relación matemática entre los giros de las articulaciones con coordenadas en el espacio del extremo del brazo. Posteriormente, se presenta el planteamiento de la cinemática directa mediante la formulación de Denavit – Hartenberg.

El tercer capítulo, diseño y ensamble del brazo robot, se inicia con el planteamiento del problema fundamentando la necesidad de contar con un brazo robótico manipulador para fines experimentales; luego se establecen los criterios de diseño y ensamble donde se analizan los grados de libertad y cada uno de los componentes del brazo; después se detallan las especificaciones de trabajo del brazo robot que involucra las dimensiones de las piezas que se pueden escanear así como los grados de giro recomendados para cada una de sus articulaciones.

El cuarto capítulo, programación para el control del brazo robot, incorpora la descripción de los software y hardware utilizados para el procesamiento de la información para desarrollar el escaneo de curvas, donde se muestran además los diagramas de flujo que buscan simplificar y hacer más entendible la programación.

El quinto capítulo, manual de uso del brazo robot, muestra los ensayos y pruebas de escaneo a curvas y se calcula error de precisión de las curvaturas de cada uno de los ensayos.

Finalmente se presentan las conclusiones del presente trabajo y las recomendaciones para el desarrollo de trabajos futuros.

Capítulo 1

Marco teórico

1.1 Antecedentes históricos

Zunt, D. (2004) en su artículo ¿Quién inventó realmente la palabra "robot" y qué significa? cuenta que la palabra "robotnik" fue dada a conocer por el dramaturgo checo Karel Čapek en 1921, en su obra R.U.R. (Robots Universales Rossum), aunque originalmente los quiso llamar laboř (del latín labor, trabajo); pero fue su hermano Josef quien le sugirió "roboti", derivado de la palabra "robota" que significa literalmente trabajo o labor y figuradamente "trabajo duro".

George Devol, inventor estadounidense, pionero de la robótica industrial, patentó en 1948, un manipulador programable que fue el germen del robot industrial. Ese mismo año Goertz R. del Argonne National Laboratory desarrollo el primer tele manipulador, con el objetivo de manipular elementos radioactivos sin riesgo para el operador; y en 1954, desarrolló el primer tele manipulador con servocontrol bilateral con uso de la tecnología electrónica y del servocontrol sustituyendo así la transmisión mecánica por eléctrica. Otro de los pioneros de la tele manipulación fue Ralph Mosher, ingeniero de la General Electric, quien en 1958 desarrolló un dispositivo denominado Handy-Man, consistente en dos brazos mecánicos tele operados mediante un maestro. En los años sesenta la industria submarina comenzó a interesarse por el uso de los tele manipuladores.

El crecimiento de la robótica en Japón aventaja en breve a los Estados Unidos gracias a Nissan, que formó la primera asociación robótica del mundo, la Asociación de Robótica industrial de Japón (JIRA) en 1971. Dos años más tarde se formó el Instituto de Robótica de América (RIA), que en 1984 cambió su nombre por el de Asociación de Industrias Robóticas, manteniendo las mismas siglas (RIA). Por su parte, Europa tuvo un despertar más tardío; en 1973 la firma sueca ASEA construyo el primer robot con accionamiento totalmente eléctrico, en 1980 se fundó la Federación Internacional de Robótica con sede en Estocolmo Suecia.

La configuración de los primeros robots respondía a las denominadas configuraciones esférica y antropomórfica, de uso especialmente valido para la manipulación. En 1982, el profesor Makino de la Universidad Yamanashi de Japón, desarrolla el concepto de robot SCARA (Selective Compliance Assembly Robot Arm) que busca un robot con un número reducido en grados de libertad (3 o 4), un costo limitado y una configuración orientada al ensamblado de piezas.

La evolución de los robots industriales desde sus principios ha sido vertiginosa y siempre estuvo orientada a funciones de manipulación. Durante los años ochenta la robótica móvil fue cobrando una importancia creciente porque pueden desplazarse por el terreno, el agua o incluso volar libremente como es el caso más reciente de los drones para aplicaciones en la industria agrícola. Así, en poco más de 30 años las investigaciones y desarrollos sobre robótica industrial han permitido que los robots tomen posiciones en casi todas las áreas productivas y tipos de industria. En pequeñas o grandes fábricas, los robots pueden sustituir al hombre en aquellas áreas repetitivas y hostiles, adaptándose inmediatamente a los cambios de producción solicitados por la demanda variable.

1.2 Definición y clasificación de los robots

1.2.1 Definición del robot industrial

La definición más comúnmente aceptada posiblemente sea la de la Asociación de Industrias Robóticas (RIA) de los Estados Unidos, según la cual un **robot industrial** es un manipulador multifuncional reprogramable, capaz de mover materias, piezas, herramientas, o dispositivos especiales, según trayectorias variables, programadas para realizar tareas diversas.

Una definición más completa es la establecida por la Asociación Francesa de Normalización (AFNOR) que define:

- **Manipulador:** mecanismo formado generalmente por elementos en serie, articulados entre sí, destinado al agarre y desplazamiento de objetos. Es multifuncional y puede ser gobernado directamente por un operador humano o mediante dispositivo lógico.
- **Robot:** manipulador automático servo controlado, reprogramable, polivalente, capaz de posicionar y orientar piezas, útiles o dispositivos especiales, siguiendo trayectoria variables reprogramables, para la ejecución de tareas variadas. Normalmente tiene la forma de uno o varios brazos terminados en una muñeca. Su unidad de control incluye un dispositivo de memoria y ocasionalmente de percepción del entorno. Normalmente su uso es el de realizar una tarea de manera cíclica, pudiéndose adaptar a otra sin cambios permanentes en su material.

Por último, la Federación Internacional de Robótica (IFR) distingue entre robot industrial de manipulación y otros robots:

- Por robot industrial de manipulación se entiende a una máquina de manipulación automática, reprogramable y multifuncional con tres o más ejes que pueden posicionar y orientar materias, piezas, herramientas o dispositivos especiales para la ejecución de trabajos diversos en las diferentes etapas de la producción industrial, ya sea en una posición fija o en movimiento. En esta definición se debe entender que la reprogramabilidad y multifunción se consigue sin modificaciones físicas del robot.

- Común en todas las definiciones anteriores es la aceptación del robot industrial como un brazo mecánico con capacidad de manipulación y que incorpora un control más o menos complejo.

Un sistema robotizado, en cambio, es un concepto más amplio; engloba todos aquellos dispositivos que realizan tareas de forma automática en sustitución de un ser humano y que pueden incorporar o no a uno o varios robots, siendo esto último lo más frecuente.

El estándar ISO 8373 (1994) define un robot industrial como un manipulador programable en tres o más ejes multipropósito, controlado automáticamente y reprogramable.

1.2.2 Clasificación del robot industrial

1. Manipuladores.

La estructura mecánica de un manipulador consta de una secuencia de cuerpos rígidos conectados por medio de articulaciones. Se caracteriza por un brazo que asegura la movilidad, una muñeca que confiere destreza, y un efector terminal que realiza la tarea requerida.

Los manipuladores son sistemas mecánicos multifuncionales, con un sencillo sistema de control que permite gobernar el movimiento de sus elementos de los siguientes modos:

- Manipulador manual: Cuando el operario controla directamente la tarea del manipulador.
- De secuencia fija: cuando se repite, de forma invariable, el proceso de trabajo preparado previamente.
- De secuencia variable: Se pueden alterar algunas características de los ciclos de trabajo

Existen muchas operaciones básicas que pueden ser realizadas óptimamente mediante manipuladores, por lo que se debe considerar seriamente el empleo de estos dispositivos, cuando las funciones de trabajo sean sencillas y repetitivas.

2. Robots de repetición o aprendizaje.

Son manipuladores que se limitan a repetir una secuencia de movimientos, previamente ejecutada por un operador humano, haciendo uso de un controlador manual o un dispositivo auxiliar. En este tipo de robots, el operario en la fase de enseñanza, se vale de una pistola de programación con diversos pulsadores o teclas, o bien, de joysticks, o bien utiliza un maniquí, o a veces, desplaza directamente la mano del robot. Los robots de aprendizaje son los más conocidos, hoy día, en los ambientes industriales y el tipo de programación que incorporan, recibe el nombre de "gestual".

3. Robots inteligentes.

Son similares a los del grupo anterior, pero también son capaces de relacionarse con el mundo que les rodea a través de sensores y tomar decisiones en tiempo real (auto programable).

4. Micro-robots.

Tienen fines educacionales, de entretenimiento o investigación. Existen numerosos robots de formación o micro-robots a un precio muy asequible y, cuya estructura y funcionamiento son similares a los de aplicación industrial.

1.2.3 Clasificación según el tipo de trayectoria

Según el manual técnico OSHA, los robots industriales se pueden programar a distancia para llevar a cabo sus operaciones requeridas con tres diferentes tipos de trayectorias:

- Ruta de punto a punto. Los robots programados y controlados de esta manera están programadas para moverse de un punto discreto a otro dentro de su área de trabajo. En el modo automático de operación, la ruta exacta adoptada por el robot puede variar ligeramente debido a las variaciones en la velocidad, geometrías de junta, y las ubicaciones espaciales de puntos.
- Trayectoria controlada. La ruta de acceso o el modo de movimiento asegura que el extremo del brazo del robot seguirá una trayectoria predecible (controlada) y una orientación tal que el robot se desplaza de un punto a otro. Las transformaciones de coordenadas necesarias para esta gestión de hardware son calculadas por ordenador del sistema de control del robot.
- Recorrido continuo. Un robot cuya trayectoria se controla mediante el almacenamiento de una gran cantidad o cerca sucesión de puntos espaciales en la memoria durante una secuencia de enseñanza es un robot controlado continuo camino. Durante este tiempo, y mientras que el robot se mueve, los puntos de coordenadas en el espacio de cada eje se controlan continuamente en una base de tiempo fijo, por ejemplo, 60 o más veces por segundo, y se coloca en la memoria del ordenador del sistema de control. Cuando el robot se coloca en el modo de funcionamiento automático, el programa se reproduce desde la memoria y se genera una trayectoria duplicada.

1.2.4 Clasificación según AFRI

La Asociación Francesa de Robótica Industrial (AFRI) clasifica los robots por tipos, según se muestra en la tabla 1.1:

Tabla 1.1. Clasificación de los robots según la AFRI

Tipo A	Manipulador con control manual o telemando.
Tipo B	Manipulador automático con ciclos preajustados; regulación mediante fines de carrera o topes; control por PLC; accionamiento neumático, eléctrico o hidráulico.
Tipo C	Robot programable con trayectoria continua o punto a punto. Carece de conocimiento sobre su entorno
Tipo D	Robot capaz de adquirir datos de su entorno, readaptando su tarea en función de estos.

Fuente: Asociación Francesa de Robótica Industrial.

1.2.5 Clasificación según IFR

La IFR (International Federation of Robotics) distingue entre cuatro tipos de robots: secuencial, de trayectoria controlable, adaptativo y tele manipulado y los clasifica por generaciones (tabla 1.2).

Tabla 1.2. Clasificación de los robots industriales por generaciones

1º Generación	Repite la tarea programada secuencialmente. No toma en cuenta las posibles alteraciones de su entorno.
1º Generación	Adquiere información limitada de su entorno y actúa en consecuencia. Puede localizar, clasificar (visión) y detectar esfuerzos y adaptar sus movimientos en consecuencia.
1º Generación	Su programación se realiza mediante el empleo de un lenguaje natural. Posee la capacidad para la planificación automática de sus tareas

Fuente: International Federation of Robotics (IFR) recuperado de www.ifr.org.

1.2.6 Clasificación según Knasel

Según el físico matemático T.M. Knasel, ilustra cinco generaciones de los robots según las características de comando/control, movilidad, empleo y fecha de madurez de cada uno, tal como se muestra en la tabla 1.3.

Tabla 1.3. Clasificación de los robots según Knasel

Generación	Nombre	Tipo de Control	Grado de movilidad	Usos más frecuentes
1 (1982)	Pick & place	Fines de carrera, aprendizaje	Ninguno	Manipulación, servicio de máquinas
2 (1984)	Servo	Servocontrol, Trayectoria continua, programación condicional.	Desplazamiento por vía	Soldadura, pintura
3 (1989)	Ensamblado	Servos de precisión, visión, tacto.	Guiado por vía	Ensamblado, Desbaldado
4 (2000)	Móvil	Sensores inteligentes	Patas, Ruedas	Construcción, Mantenimiento
5 (2010)	Especiales	Controlados con técnicas de IA	Andante, Saltarín	Militar, Espacial

Fuente: Knasel T.(1986). Mobile Robots-state of the art review. Elsevier Journal..

1.3 Morfología de los Manipuladores

1.3.1 Grados de libertad

Los grados de libertad son el número de parámetros independientes que fijan la situación del órgano terminal. El número de grados de libertad suele coincidir con el número de eslabones de la cadena cinemática y definen los posibles movimientos básicos e independientes de los robots (giratorios y de desplazamiento). Son los parámetros que se precisan para determinar la posición y la orientación del elemento terminal del manipulador. Un mayor número de grados de libertad conlleva un aumento de la flexibilidad en el posicionamiento del elemento terminal. Las tareas más sencillas y con movimientos más limitados, como las de la pintura y palatización, suelen exigir 4 o 5 grados de libertad. La mayoría de las aplicaciones industriales requieren 6 grados de libertad, como las de soldadura y mecanizado, mientras que otras más complejas reciben un número mayor, tal es el caso en las labores de montaje.

1.3.2 Estructuras básicas

La estructura básica de un manipulador consiste en un brazo compuesto por elementos con articulaciones entre ellos. En el último enlace se coloca un órgano terminal o efector final tal como una pinza o dispositivo especial para realizar operaciones.

En primer lugar los brazos de un robot manipulador son las estructuras más utilizadas y tienen diferentes propiedades en cuanto a espacio de trabajo y accesibilidad a posiciones determinadas. En la figura 1.1 se muestran cuatro configuraciones básicas

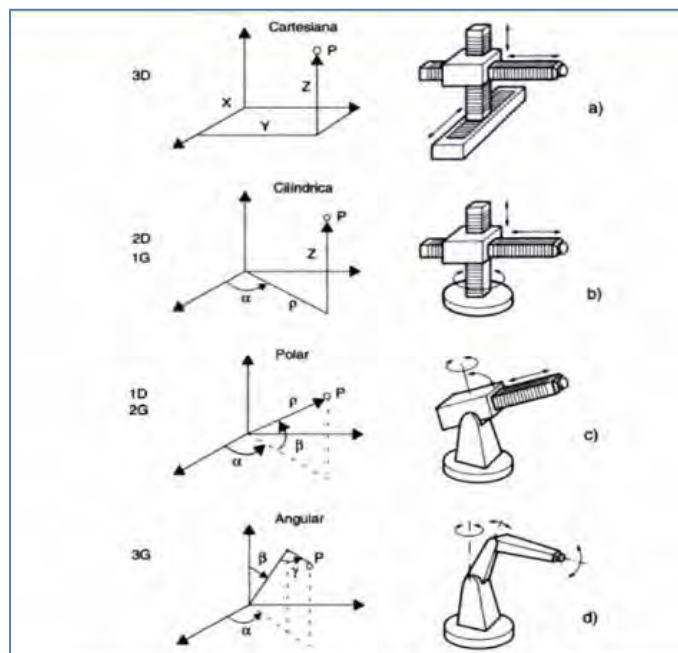


Figura 1.1. Configuraciones básicas de robots manipuladores industriales.

Fuente: Ferraté et al. (1986).

El espacio de trabajo es el conjunto de puntos en los que puede situarse el efector del manipulador. Todos los puntos del espacio de trabajo no tienen la misma accesibilidad. Los puntos de accesibilidad mínima son los de las superficies que delimitan el espacio de trabajo y puede llegarse a ellos con una única orientación.

- La configuración cartesiana tiene tres articulaciones prismáticas y es bastante usual en estructuras industriales. La especificación de un punto se efectúa mediante coordenadas cartesianas (x, y, z). Los valores que deben tomar las variables articulares corresponden directamente a las coordenadas que toma el extremo del brazo; sin embargo, esta configuración no resulta adecuada para acceder a puntos situados en espacios relativamente cerrados y su volumen de trabajo es pequeño.
- La configuración cilíndrica tiene dos articulaciones prismáticas y una de rotación. La primera articulación es normalmente de rotación y la posición se especifica en coordenadas cilíndricas.
- La configuración polar cuenta con dos articulaciones rotacionales y una lineal.
- La configuración angular (o de brazo articulado) posee tres articulaciones rotacionales.

Además de estas cuatro configuraciones básicas se han desarrollado combinaciones entre ellas, destacándose la “configuración mixta” tipo SCARA (*Selective Compliance Assembly Robot Arm*) y la “configuración en paralelo” que posee brazos con articulaciones prismáticas o rotacionales concurrentes.

Los robots SCARA y antropomórfico presentan una estructura donde se combinan articulaciones prismáticas y angulares de tal forma que puedan cumplir ciertas características de rigidez y precisión para la realización de alguna tarea. El manipulador SCARA es adecuado para la manipulación de pequeños objetos, mientras que el manipulador antropomórfico, en su similitud con el brazo humano, presenta sólo articulaciones angulares que generalmente son accionadas por motores eléctricos.

1.3.3 Articulaciones

La estructura fundamental de un manipulador es la cadena cinemática en serie o abierta, es decir cuando existe una cadena cinemática con una secuencia de enlaces que conecten los dos extremos de la cadena; caso contrario a la cadena cinemática cerrada, donde la secuencia de enlaces forma un bucle.

La movilidad de los manipuladores está garantizada por la presencia de articulaciones. En una cadena cinemática abierta, cada articulación proporciona la estructura con un solo grado de libertad. Dicha junta o unión crea un movimiento de rotación relativa entre dos enlaces siendo necesaria la distribución adecuada de articulaciones para ejecutar una determinada tarea. Por otra parte, en una cadena cinemática cerrada, el número de grados de libertad es menor que el número de articulaciones, debido a las limitaciones impuestas por el bucle.

Con respecto a las articulaciones, se pueden distinguir dos tipos:

- Articulaciones prismáticas: Llamada también junta deslizante, ya que posibilita a un eslabón del brazo deslizarse en línea recta sobre otro
- Articulaciones giratorias: Que por su mismo nombre, permite un giro como una puerta y su bisagra.

1.3.4 Acondicionamiento de un brazo manipulador para lectura de posición

Para que un brazo manipulador pueda tomar lectura de posición debe contar con un circuito de acondicionamiento con sensores instalados en cada articulación; en este caso se incorporan “encoders” ópticos incrementales. Estos sensores generan tres señales digitales A, B y Z tal como se muestra en la figura 2. Dependiendo de la resolución del encoder, las señales A y B generan una cantidad de pulsos digitales de revolución, mientras que la señal Z siempre genera un pulso por revolución indicando que el motor ha girado 360° . Con la lectura de la señal A se puede obtener la información de posición y velocidad del motor y con la lectura de la señal B se obtiene la información del sentido del giro del motor.

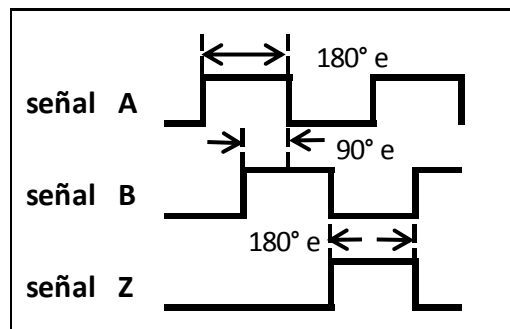


Figura 1.2. Representación gráfica de las señales incrementales A, B y Z.

Fuente: Madrid E. (2007). Octavo Congreso Iberoamericano de Ingeniería Mecánica. Cusco.

Los “encoders” son una clase de sensor de posición utilizado para la adquisición de datos.

1.4 Sistema de adquisición de datos

Los sistemas de instrumentación para la adquisición de datos tienden a componerse de varios componentes para realizar una medida y visualizar o registrar el resultado (figura 1.3). Estos componentes son dispositivos de entrada que recibe una señal y la convierte en un formato adecuado para la siguiente componente que acondiciona la señal para ser procesada posteriormente. Los dispositivos de entrada se conocen como sensores, detectores, captadores, sondas y transductores.

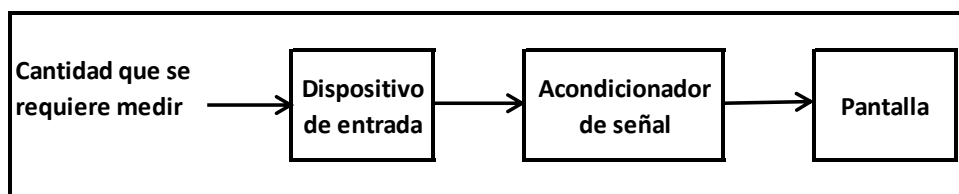


Figura 1.3. Forma general de un sistema de medida.

Fuente: Boltom W. (1995) Mediciones y pruebas eléctricas y electrónicas: Instrumento automáticos.

1.4.1 Transductores

Los términos sensor y “transductor” se suelen aceptar como sinónimos; aunque si hubiera que hacer alguna distinción, el término transductor es quizás más amplio, incluyendo una parte sensible o “captador” propiamente dicho y algún tipo de circuito de acondicionamiento de la señal detectada.

Un transductor es un dispositivo capaz de convertir el valor de una magnitud física en una señal eléctrica codificada, ya sea de forma analógica o digital. No todos los transductores tienen por qué dar salida en forma de señal eléctrica. Como ejemplo puede valer el caso de un termómetro basado en la diferencia de dilatación donde la temperatura se convierte directamente en desplazamiento de una aguja indicadora. Sin embargo, el término transductor suele asociarse a dispositivos cuya salida es alguna magnitud eléctrica o magnética. Y pueden ser conectados a autómatas programables a través de las interfaces adecuadas.

En general, los transductores tienen la siguiente estructura (figura 1.4):

- Elemento sensor o capturador, que convierte las variaciones de una magnitud física en variaciones de una magnitud eléctrica o magnética, que se denomina señal.
- Bloque de tratamiento de señal, filtra, amplifica, linealiza y modifica la señal obtenida en el capturador.
- Etapa de salida, que comprende los amplificadores, interruptores, convertidores de código, transmisores y todas aquellas partes que adaptan la señal a las necesidades de la carga exterior.

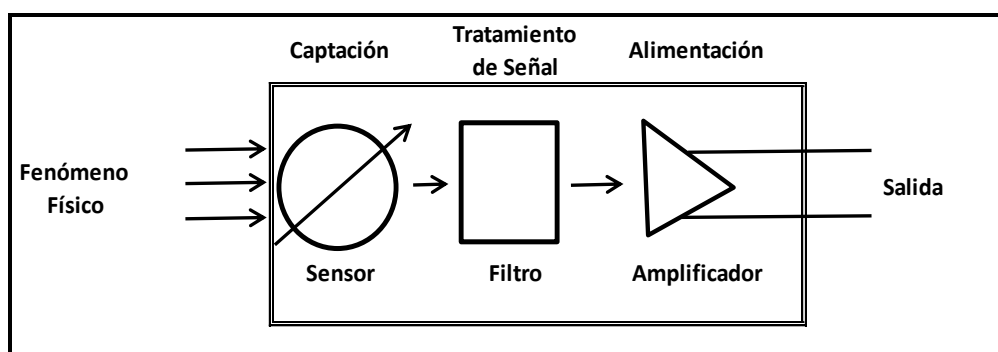


Figura 1.4. Estructura genérica de transductor.

Fuente: Balcells J. y Romeral J. (1997). Autómatas programables.

1.4.1.1 Clasificación

Hay varias clasificaciones de los transductores, atendiendo a diversos puntos de vista se pueden clasificar de la siguiente manera:

1. Según el tipo de señal de salida:

- a) Analógicos: aquellos que dan como salida un valor de tensión o corriente variable del campo de medida. Es frecuente para este tipo de transductores que incluyan una etapa de salida para suministrar de 0-10 voltios o 4-20 micro amperios.
- b) Digitales: son aquellos que dan como salida una señal codificada en forma de pulsos.
- c) Todo-nada: indican únicamente cuando la variable detectada rebasa un cierto umbral o límite, en el caso límite estos sensores codifican sólo dos estados.

Otro criterio de clasificación según la señal de salida es:

- a) Sensores pasivos: se basan en la modificación de la impedancia eléctrica o magnética de un material bajo determinadas condiciones físicas o químicas (resistencia, capacidad, inductancia, reluctancia, etc.). Este tipo de sensores provoca cambios de tensión o corriente en un circuito, los cuales son recogidos por el circuito de interfaz.
- b) Sensores activos son generadores eléctricos, generalmente de pequeña señal; por ello no necesitan alimentación exterior para funcionar, aunque suelen necesitarla para amplificar la débil señal del captador.

2. Según la magnitud física a detectar

En cuanto a la naturaleza de la magnitud física a detectar, existe una gran variedad de sensores. En la tabla 1.4 se muestra un resumen de los más frecuentes utilizados en los automáticos industriales. Se puede observar en la columna encabezada por “Transductor” aparece a veces el nombre del elemento captador de dicho transductor, sobre todo en los casos de medición indirecta.

Tabla 1.4. Transductores de diversas magnitudes físicas.

Magnitud	Transductor	Característica
Posición lineal y angular	Potenciómetro Encoder Sensor Hall	Analógica Digital Digital
Desplazamiento y deformación	Transformador diferencial de variación lineal Galga extensiométrica Magnetoestrictivos Magnetoresistivos LVDT	Analógica Analógica A/D Analógica Analógica
Velocidad lineal y angular	Dinamo tacométrica Encoder Detector inductivo Servo-inclinómetros RVDT Giróscopo	Analógica Digital Digital A/D Analógica
Aceleración	Acelerómetro Servo-accelerómetros	Analógico
Fuerza y par (deformación)	Galga extensiométrica Triaxiales	Analógico A/D
Presión	Membranas Piezoeléctricos Manómetros Digitales	Analógica Analógica Digital
Caudal	Turbina Magnético	Analógica Analógica
Temperatura	Termopar RTD Termistor NTC Termistor PTC [Bimetal - Termostato	Analógica Analógica Analógica Analógica I/O
Sensores de presencia	Inductivos Capacitivos Ópticos	I/O I/O I/O y Analógica
Sensores táctiles	Matriz de contactos Piel artificial	I/O Analógica
Visión artificial	Cámaras de video Cámaras CCD o CMOS	Procesamiento digital Procesamiento
Sensor de proximidad	Sensor final de carrera Sensor capacitivo Sensor inductivo Sensor fotoeléctrico	Analógica Analógica Analógica
Sensor acústico Sensores de acidez	micrófono ISFET	Analógica
Sensor de luz	fotodiodo Fotorresistencia Fototransistor Célula fotoeléctrica	Analógica Analógica Analógica Analógica
Sensores captura de movimiento	Sensores inerciales	

Fuente: Balcells J. y Romeral J. (1997). Autómatas programables.

1.4.1.2. Características dinámicas de los transductores

La mayor parte de los transductores tienen un comportamiento dinámico. Las principales características son:

1. Velocidad de respuesta: mide la capacidad de un transductor para que la señal de salida siga sin retraso las variaciones de la señal de entrada. La forma de cuantificar este parámetro es en base de una a más constantes de tiempo, que suelen obtenerse de la respuesta al escalón. Los parámetros más relevantes empleados en la definición de la velocidad de respuesta son los siguientes:
 - Tiempo de retardo, es el tiempo transcurrido desde la aplicación del escalón de entrada hasta que la salida alcanza el 10% de su valor permanente.
 - Tiempo de subida, es el tiempo transcurrido desde que la salida alcanza el 10% de su valor permanente hasta que llega por primera vez al 90% de dicho valor.
 - Tiempo de establecimiento al 99%, es el tiempo transcurrido desde la aplicación de un escalón de entrada hasta la respuesta alcanza el régimen permanente, con una tolerancia de $\pm 1\%$.
 - Constantes de tiempo, para un transductor con respuesta de primer orden (una sola constante de tiempo dominante) se puede determinar la constante de tiempo a base de medir el tiempo empleado para que la salida alcance el 63% de su valor de régimen permanente, cuando a la entrada se aplica un cambio de escalón.
2. Respuesta frecuencial: está directamente relacionada con la velocidad de respuesta, y es la relación entre la sensibilidad y la frecuencia cuando la entrada es una excitación senoidal. Se suele indicar gráficamente mediante un diagrama de Bode.
3. Estabilidad y derivas, son características que indican la desviación de salida del sensor al variar ciertos parámetros exteriores distintos del que se pretende medir, tales como condiciones ambientales, alimentación u otras perturbaciones.

1.4.1.3. Transductores de posición

Los transductores de posición permiten medir la distancia de un objeto respecto a un punto de referencia o simplemente detectar la presencia de un objeto a cierta distancia. Su capacidad de medida permite establecer una división en grupos:

- Detectores de presencia o proximidad, son sensores de posición todo o nada que entregan una señal binaria que informa la existencia o no de un objeto ante el detector. El más elemental de estos sensores es quizás el conocido interruptor final de carrera por contacto mecánico.

- Medidores de distancia o posición, entregan una señal analógica o digital que permite determinar la posición lineal o angular respecto a un punto o eje de referencia.
- Transductores de pequeñas deformaciones, se trata de posición especialmente diseñados para detectar pequeñas deformaciones o movimientos. Muchas veces se emplean adosados a piezas elásticas o con palpadores como transductores indirectos de fuerza o de par.

1.4.1.4. Captadores de proximidad

Los captadores de proximidad son sensores todo o nada y pueden ser inductivos, capacitivos, óptico y ultrasónicos. Respecto a los sensores ópticos, utilizados en este trabajo, se puede decir que emplean fotocélulas como elementos de detección. Algunos tipos disponen de un cabezal que incorpora un emisor de luz y la fotocélula de detección, actuando por reflexión y detección del haz de luz reflejado sobre el objeto que se pretende detectar.

1.4.2. Encoders.

Los encoders son dispositivos formados por un rotor con uno o varios grupos de bandas y por una serie de captadores ópticos alojados en el estator que detectan la presencia o no de banda opaca frente a ellos. Existen dos tipos de encoders:

- Incrementales que dan un número determinados de impulsos por vuelta y requieren un contador para determinar la posición a partir de un origen de referencia.
- Absolutos que disponen de varias bandas en el rotor ordenadas según un código binario.

1.4.2.1. Encoders incrementales o relativos.

Los encoders incrementales suelen tener una sola banda de marcas transparentes y opacas repartidas a lo largo del disco rotórico y separadas por un paso (p), tal como muestra la figura 5.

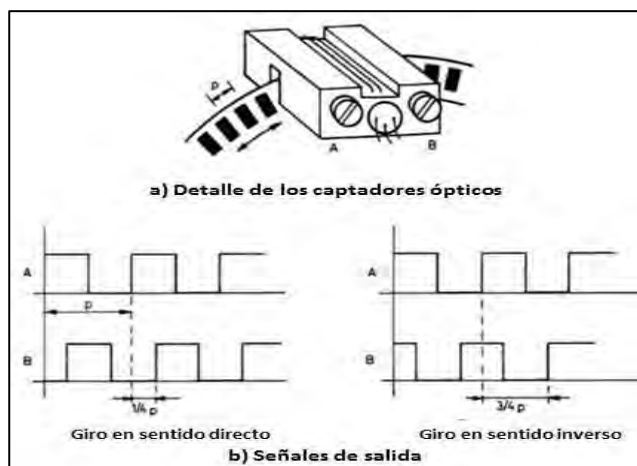


Figura 1.5. Encoder incremental.

Fuente: Balcells J (1997). Autómatas programables.

En el estator suelen disponer de dos pasos de $\frac{1}{4}$. Al girar el rotor, cada par óptico genera una señal cuadrada. El decalaje de $\frac{1}{4}$ de división de los captadores hace que las señales cuadradas de salida tengan entre sí un desfase de $\frac{1}{4}$ de periodo cuando el motor giran en sentido contrario, lo cual se utiliza para discriminar el sentido de giro.

Un simple sistema lógico permite determinar desplazamientos a partir de un origen, a base de contar los impulsos de un canal y determinar el sentido giro a partir del desfase entre las señales de los dos canales, según se muestra en la figura 1.6.

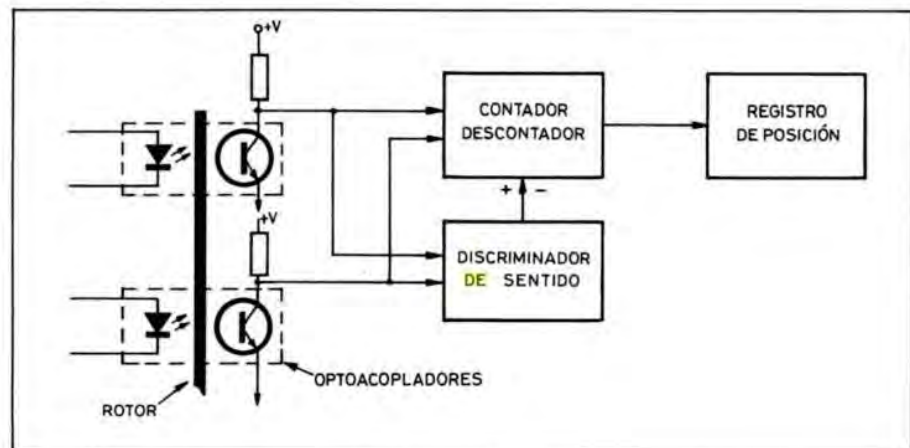


Figura 1.6. Discriminador de posición y sentido de giro.

Fuente: Balcells J. (1997). Autómatas programables.

Algunos encoders incrementales disponen de un canal adicional, que proporciona un impulso por revolución. La lógica de control puede utilizar esta señal para implementar un contador devueltas y otro para fracciones de vueltas.

La resolución del encoder dependerá del número(N) de divisiones del rotor o, lo que es lo mismo, del número de impulsos por revolución. La resolución expresada en grados vale: $360/N$.

Capítulo 2

Fundamentos de Robótica

2.1 Herramientas matemáticas para la localización espacial

La manipulación de piezas llevada a cabo por un robot implica el movimiento espacial de su extremo, por ello es necesario conocer la posición y orientación de ésta con respecto a la base del robot. Existe una serie de herramientas matemáticas que permiten especificar la posición y orientación en el espacio de piezas, herramientas y, en general, de cualquier objeto.

2.1.1 Representación de la posición

Para localizar un cuerpo rígido en el espacio es necesario contar con una herramienta que permita la localización espacial de sus puntos. En un plano el posicionamiento tiene dos grados de libertad, y por tanto la posición de un punto viene definida por dos componentes independientes. En el caso de un espacio tridimensional será necesario emplear tres componentes.

La posición de un punto puede definirse por coordenadas cartesianas, coordenadas polares para dos dimensiones, coordenadas cilíndricas y esféricas para espacios de tres dimensiones.

- El sistema cartesiano de referencia se define mediante ejes perpendiculares entre sí con un origen definido. En el caso de trabajar en el plano (2 dimensiones), el sistema de referencia OXY correspondiente queda definido por dos vectores coordenados OX y OY perpendiculares entre sí con un punto de intersección común O. Si se trabaja en el espacio (tres dimensiones), el sistema cartesiano OXYZ está compuesto por una terna ortogonal de vectores coordenados OX, OY y OZ (figura 2.1).

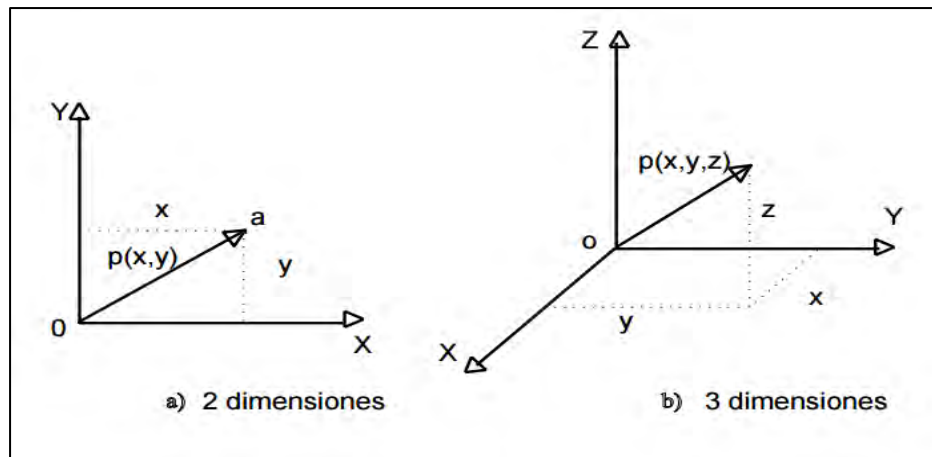


Figura 2.1. Representación de un vector en coordenadas cartesianas.

Fuente: Barrientos A., Peñin L., Balaguer C. y Aracil R. (2007).
Fundamentos de robótica.

- Las coordenadas polares y cilíndricas también caracterizan la localización de un punto o vector “p” respecto a un sistema de ejes cartesianos de referencia OXY utilizando las denominadas coordenadas polares $p(r,\theta)$. En esta representación, r representa la distancia desde el origen O del sistema hasta el extremo del vector p , mientras que θ es el ángulo que forma el vector p con el eje OX. En el caso de trabajar en tres dimensiones, un vector p podrá expresarse con respecto a un sistema de referencia OXYZ, mediante las coordenadas cilíndricas $p(r,\theta,z)$. Las componentes r y θ tienen el mismo significado que en el caso de coordenadas polares, aplicado el razonamiento sobre el plano OXY, mientras que la componente z expresa la proyección sobre el eje OZ del vector p (figura 2.2).

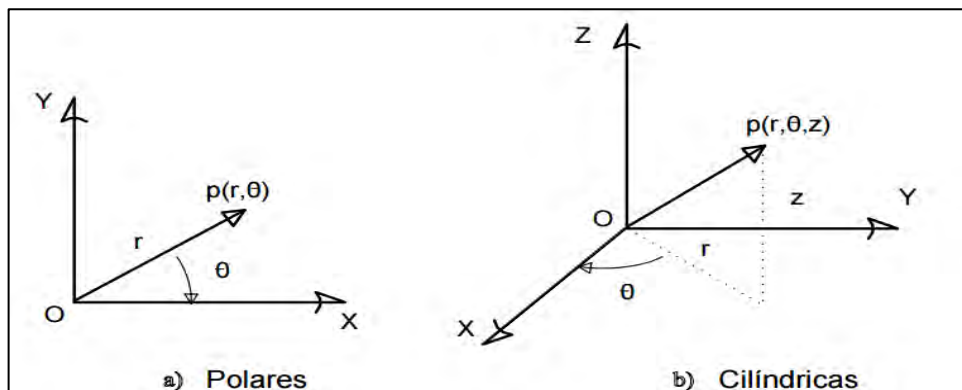


Figura 2.2. Representación de un vector en coordenadas polares y cilíndricas.

Fuente: Barrientos A., Peñin L., Balaguer C. y Aracil R. (2007).
Fundamentos de robótica.

- Las coordenadas esféricas se utilizan para la localización de un vector en un espacio de tres dimensiones, mediante el sistema de referencia OXYZ, el vector p tendrá como coordenadas esféricas (r,θ,Φ) , donde la componente r es la distancia desde el origen O hasta el extremo del vector p ; la componente θ es el ángulo formado por el vector p con el eje OZ (figura 2.3).

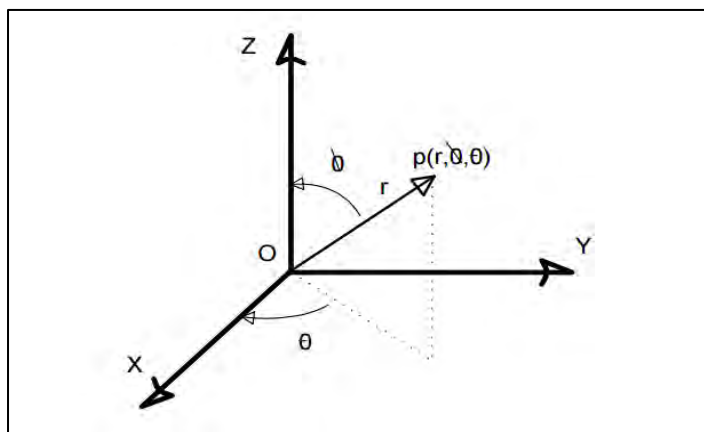


Figura 2.3. Representación de un vector en coordenadas esféricas.

Fuente: Barrientos A., Peñín L., Balaguer C. y Aracil R. (2007).
Fundamentos de robótica.

2.1.2 Representación de la orientación

Un punto queda totalmente definido en el espacio a través de los datos de su posición; sin embargo, para el caso de un sólido, es necesario además definir cuál es su orientación con respecto a un sistema de referencia.

Una orientación en el espacio tridimensional viene definida por tres grados de libertad o tres componentes linealmente independientes. Para poder describir de forma sencilla la orientación de un objeto, respecto a un sistema de referencia, es habitual asignar solidariamente al objeto un nuevo sistema, y después estudiar la relación espacial existente entre los dos sistemas. De forma general, esta relación vendrá dada por la posición y orientación del sistema asociado al objeto respecto al de referencia.

Para el análisis de los distintos métodos de representar orientaciones se supondrá que ambos sistemas coinciden en el origen, y que por tanto no existe cambio alguno de posición entre ellos.

2.1.2.1. Matrices de rotación

Las matrices de rotación son el método más extendido para la descripción de orientaciones, debido principalmente a la comodidad que proporciona el uso del álgebra matricial.

En la figura 2.4 el sistema OXY es el de referencia fija, y el sistema OUV es el móvil solidario al objeto. Los vectores unitarios del sistema OXY son $\hat{i}_x, \hat{j}_x$, en tanto que los del sistema OUV son $\hat{i}_u, \hat{j}_v$.

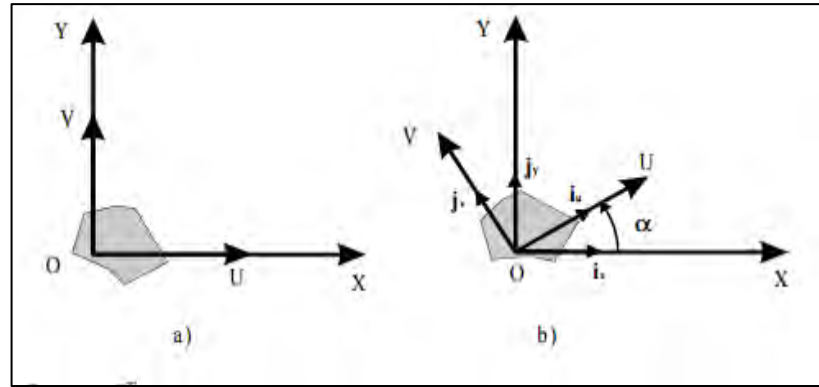


Figura 2.4. Orientación de un sistema OUV respecto a otro OXY en un plano.

Fuente: Barrientos A., Peñín L., Balaguer C. y Aracil R. (2007). Fundamentos de robótica.

La representación en ambos sistemas del vector p en el plano, se registra de la siguiente manera:

$$p_{x,y} = [p_x, p_y]^T = p_x \cdot i_x + p_y \cdot j_y \quad \text{Ec. 2.1}$$

$$p_{u,v} = [p_u, p_v]^T = p_u \cdot i_u + p_v \cdot j_v \quad \text{Ec. 2.2}$$

Después de realizar una serie de transformaciones:

$$\begin{bmatrix} p_x \\ p_y \end{bmatrix} = R \begin{bmatrix} p_u \\ p_v \end{bmatrix} \quad \text{Ec. 2.3}$$

De donde:

$$R = \begin{bmatrix} i_x i_u & i_x j_v \\ j_y i_u & j_y j_v \end{bmatrix} \quad \text{Ec. 2.4}$$

Es la llamada matriz de rotación que define la orientación del sistema OUV con respecto al sistema OXY, y que sirve para transformar las coordenadas de un vector en un sistema a las del otro. También recibe el nombre de matriz de cosenos directores. Es fácil de comprobar que se trata de una matriz ortonormal, tal que:

$$R^{-1} = R^T \quad \text{Ec. 2.5}$$

En el caso de dos dimensiones, la orientación viene definida por un único parámetro independiente. Si se considera la posición relativa del sistema OUV girado un ángulo α sobre el OXY tras realizar los correspondientes productos escalares, la matriz R será de la forma:

$$R = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix} \quad \text{Ec. 2.6}$$

Para el caso en que $\alpha = 0$, en el que los ejes coordenados de ambos sistemas coinciden, la matriz R corresponderá a la matriz unitaria.

En un espacio tridimensional, el razonamiento a seguir es similar. Supónganse los sistemas OXYZ y OUVW, coincidentes en el origen, siendo el OXYZ el sistema de referencia fijo, y el OUVW el solidario al objeto cuya orientación se desea definir. Los vectores unitarios del sistema OXYZ serán i_x, j_y, k_z , mientras que los del OUVW serán i_u, j_v, k_w . Un vector p del espacio podrá ser referido a cualquiera de los sistemas de la siguiente manera:

$$P_{uvz} = [P_u, P_v, P_w]^T = p_u \cdot i_u + p_v \cdot j_v + p_w \cdot k_w \quad \text{Ec. 2.7}$$

$$P_{uvz} = [P_x, P_y, P_z]^T = p_x \cdot i_x + p_y \cdot j_y + p_z \cdot k_z \quad \text{Ec. 2.8}$$

Y al igual que en dos dimensiones, se puede obtener la siguiente equivalencia:

$$\begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = R \begin{bmatrix} p_u \\ p_v \\ p_w \end{bmatrix} \quad \text{Ec. 2.9}$$

Donde:

$$R = \begin{bmatrix} i_x i_u & i_x j_v & i_x k_w \\ j_y i_u & j_y j_v & j_y k_w \\ k_z i_u & k_z j_v & k_z k_w \end{bmatrix} \quad \text{Ec. 2.10}$$

Es la matriz de rotación que define la orientación del sistema OUVW con respecto al sistema OXYZ. Al igual que en dos dimensiones, también recibe el nombre de matriz de cosenos directores y se trata de una matriz ortonormal, tal que la inversa de la matriz R es igual a su traspuesta (Ec. 2.5).

La principal utilidad de esta matriz de rotación corresponde a la representación de la orientación de sistemas girados únicamente sobre uno de los ejes principales del sistema de referencia (figura 2.5).

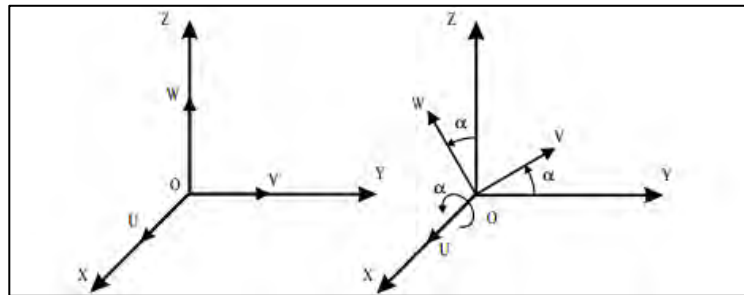


Figura 2.5. Sistema de referencia OXYZ y solidario objeto OUVW.

Fuente: Barrientos A., Peñín L., Balaguer C. y Aracil R. (2007). Fundamentos de robótica.

De la figura anterior, la orientación del sistema OUVW, con el eje OU coincidente con el eje OX, vendrá representada mediante la matriz:

$$R(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\text{sen}\alpha \\ 0 & \text{sen}\alpha & \cos\alpha \end{bmatrix} \quad \text{Ec. 2.11}$$

La orientación del sistema OUVW, con el eje OV coincidente con el eje OY, vendrá representada mediante la matriz:

$$R(y, \phi) = \begin{bmatrix} \cos\phi & 0 & \text{sen}\phi \\ 0 & 1 & 0 \\ -\text{sen}\phi & 0 & \cos\phi \end{bmatrix} \quad \text{Ec. 2.12}$$

La orientación del sistema OUVW, con el eje OW coincidente con el eje OZ, vendrá representada mediante la matriz:

$$R(z, \theta) = \begin{bmatrix} \cos\theta & -\text{sen}\theta & 0 \\ \text{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.13}$$

Viéndose más claramente en la figura 2.6.

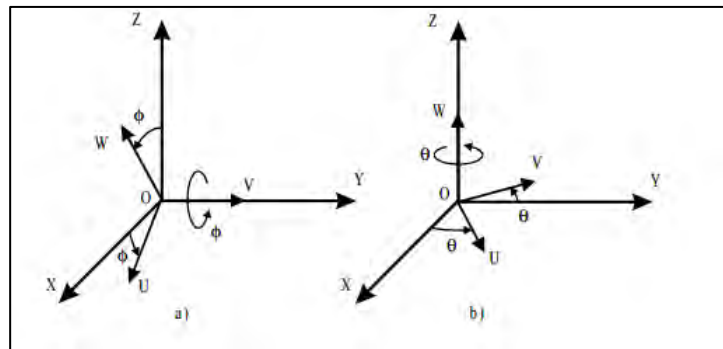


Figura 2.6. Rotación del sistema OUVW con respecto a los ejes OY y OZ.

Fuente: Barrientos A., Peñin L., Balaguer C. y Aracil R. (2007). Fundamentos de robótica.

Estas tres matrices, se denominan matrices básicas de rotación de un sistema espacial de tres dimensiones.

Composiciones de rotaciones: Las matrices de rotación pueden componerse para expresar la aplicación continua de varias rotaciones. Así, si al sistema OUVW se le aplica una rotación de ángulo α sobre OX, seguida de una rotación de ángulo Φ sobre OY y de una rotación de ángulo θ sobre OZ, la rotación global puede expresarse como:

$$T = R(z, \theta) \cdot R(y, \phi) \cdot R(x, \alpha) \quad \text{Ec. 2.14}$$

$$T = \begin{bmatrix} C & -S & 0 \\ S & C & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C\phi & 0 & S\phi \\ 0 & 1 & 0 \\ -S\phi & 0 & C\phi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha & C\alpha \end{bmatrix} \quad \text{Ec. 2.15}$$

$$T = \begin{bmatrix} C & C\phi & -S & C\alpha + C & S\phi S\alpha & S & S\alpha + C & S\phi S\alpha\phi \\ S & C\phi & C & C\alpha + S & S\phi S\alpha & -C & S\alpha + S & S\phi C\alpha \\ -S\phi & & & C\phi S\alpha & & & C\phi C\alpha & \end{bmatrix} \quad \text{Ec. 2.16}$$

Donde C expresa $\cos \theta$ y S expresa $\sin \theta$.

Es importante considerar el orden en que se realizan las rotaciones, pues el producto de matrices no es conmutativo. Así, si la rotación se hiciera primero un ángulo θ sobre OZ, seguida de una rotación de ángulo ϕ sobre OY, para finalizar con otra rotación de ángulo α sobre OX, la rotación global vendría expresada por:

$$T = R(x, \alpha) \cdot R(y, \phi) \cdot R(z, \theta) \quad \text{Ec. 2.17}$$

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & C\alpha & -S\alpha \\ 0 & S\alpha & C\alpha \end{bmatrix} \begin{bmatrix} C\phi & 0 & S\phi \\ 0 & 1 & 0 \\ -S\phi & 0 & C\phi \end{bmatrix} \begin{bmatrix} C & -S & 0 \\ S & C & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.18}$$

$$T = \begin{bmatrix} C\phi C & -C\phi S & S\phi \\ S\alpha S\phi C + C\alpha S & -S\alpha S\phi S + C\alpha C & -S\alpha C\phi \\ -C\alpha S\phi C + S\alpha S & C\alpha S\phi S + S\alpha C & C\alpha C\phi \end{bmatrix} \quad \text{Ec. 2.19}$$

2.1.2.2. Matrices de transformación homogénea

Para una representación conjunta de la posición y la orientación (localización) se introducen las denominadas coordenadas homogéneas. Coordenadas y matrices homogéneas: La representación mediante coordenadas homogéneas de la localización de sólidos en un espacio n-dimensional se realiza a través de coordenadas de un espacio (n+1)-dimensional. Es decir, un espacio n-dimensional se encuentra representado en coordenadas homogéneas por (n+1) dimensiones, de tal forma que un vector $p(x,y,z)$ vendrá representado por $p(wx,wy,wz,w)$, donde w tiene un valor arbitrario y representa un factor de escala. De forma general, un vector $p = ai + bj + ck$, donde i, j, k son los vectores unitarios de los ejes OX, OY y OZ del sistema de referencia OXYZ, se representa en coordenadas homogéneas mediante el vector columna:

$$p = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} aw \\ bw \\ cw \\ w \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \\ 1 \end{bmatrix} \quad \text{Ec. 2.20}$$

A partir de la definición de las coordenadas homogéneas surge inmediatamente el concepto de matriz de transformación homogénea. Se define como matriz de transformación homogénea T a una matriz de dimensión 4x4 que representa la transformación de un vector de coordenadas homogéneas de un sistema de coordenadas a otro.

$$T = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ f_{1 \times 3} & w_{1 \times 1} \end{bmatrix} = R = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ \text{Perspectiva} & \text{Escala} \end{bmatrix} \quad \text{Ec. 2.21}$$

Se puede considerar que una matriz homogénea se haya compuesta por cuatro submatrices de distinto tamaño: una submatriz $R_{3 \times 3}$ que corresponde a una matriz de rotación; una submatriz $p_{3 \times 1}$ que corresponde al vector de traslación; una submatriz $f_{1 \times 3}$ que representa una transformación de perspectiva y una submatriz $w_{1 \times 1}$ que representa un escalado global. En robótica generalmente sólo interesará conocer el valor de $R_{3 \times 3}$ y de $p_{3 \times 1}$, considerándose las componentes de $f_{1 \times 3}$ nulas y la de $w_{1 \times 1}$ la unidad, aunque más adelante se estudia su utilidad en otros campos. Al tratarse de una matriz 4x4, los vectores sobre los que se aplique deberán contar con dimensiones, que serán las coordenadas homogéneas del vector tridimensional de que se trate.

Aplicación de las matrices homogéneas: Si como se ha mencionado, se considera la transformación de perspectiva nula y el escalado global unitario, la matriz homogénea T resultará ser de la siguiente forma:

$$T = \begin{bmatrix} R_{3 \times 3} & p_{3 \times 1} \\ 0 & 1 \end{bmatrix} = R = \begin{bmatrix} \text{Rotación} & \text{Traslación} \\ 1 & 1 \end{bmatrix} \quad \text{Ec. 2.22}$$

Que representa la orientación y posición de un sistema $O'UVW$ rotado y trasladado con respecto al sistema de referencia $OXYZ$. Esta matriz sirve para conocer las coordenadas (r_x, r_y, r_z) del vector r en el sistema $OXYZ$ a partir de sus coordenadas (r_u, r_v, r_w) en el sistema $O'XYZ$:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} = T \begin{bmatrix} r_u \\ r_v \\ r_w \\ 1 \end{bmatrix} \quad \text{Ec. 2.23}$$

También se puede utilizar para expresar la rotación y traslación de un vector respecto de un sistema de referencia fijo $OXYZ$, de tal manera que un vector $rx y z$ rotado según $R_{3 \times 3}$ y trasladado según $p_{3 \times 1}$ se convierte en el vector $r'xyz$ dado por:

$$\begin{bmatrix} r'_x \\ r'_y \\ r'_z \\ 1 \end{bmatrix} = T \begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} \quad \text{Ec. 2.24}$$

En resumen, una matriz de transformación homogénea se puede aplicar para:

- Representar la posición y orientación de un sistema girado y trasladado $O'UVW$ con respecto a un sistema fijo de referencia

OXYZ, que es lo mismo que representar una rotación y traslación realizada sobre un sistema de referencia.

- Transformar un vector expresado en coordenadas con respecto a un sistema O'UVW, a su expresión en coordenadas del sistema de referencia OXYZ.
- Rotar y trasladar un vector con respecto a un sistema de referencia fijo OXYZ.

Se hace notar que se utilizan coordenadas homogéneas con factor de escalado la unidad, y que por tanto los vectores que intervienen en las transformaciones han de poseer cuatro componentes. Por comodidad, se elige el factor de escalado $w = 1$.

A continuación se analiza con detalle el empleo de las matrices homogéneas como herramienta para representar la localización de objetos en el espacio tridimensional, así como para realizar proyecciones y escalados.

Aplicación de las homogéneas

Traslación:

Supóngase que el sistema O'UVW únicamente se encuentra trasladado un vector $p = p_x i + p_y j + p_z k$ con respecto al sistema OXYZ. La matriz T entonces corresponderá a una matriz homogénea de traslación:

$$T(p) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.25}$$

Que es la denominada matriz básica de traslación.

Un vector cualquiera r , representado en el sistema O'UVW por r_{uvw} , tendrá como componentes del vector con respecto al sistema OXYZ:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_u \\ r_v \\ r_w \\ 1 \end{bmatrix} = \begin{bmatrix} r_u + p_x \\ r_v + p_y \\ r_w + p_z \\ 1 \end{bmatrix} \quad \text{Ec. 2.26}$$

Y a su vez, un vector r_{xyz} desplazado según T tendrá como componentes r'_{xyz} :

$$\begin{bmatrix} r'_x \\ r'_y \\ r'_z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} = \begin{bmatrix} r_x + p_x \\ r_y + p_y \\ r_z + p_z \\ 1 \end{bmatrix} \quad \text{Ec. 2.27}$$

Rotación:

Supóngase ahora que el sistema O'UVW sólo se encuentra rotado con respecto al sistema OXYZ. La sub-matriz de rotación $R_{3 \times 3}$ será la que defina la rotación, y se corresponde al tipo matriz de rotación presentada en el argumento de matrices de rotación. De igual forma que se hacía allí, se pueden definir tres matrices homogéneas básicas de rotación según se realice ésta en cada uno de los tres ejes coordenados OX, OY y OZ del sistema de referencia OXYZ:

$$T(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.28}$$

$$T(y, \phi) = \begin{bmatrix} \cos \phi & 0 & \sin \phi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \phi & 0 & \cos \phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.29}$$

$$T(z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.30}$$

Un vector cualquiera r , representado en el sistema girado O'UVW por r_{uvw} , tendrá como componentes (r_x, r_y, r_z) en el sistema OXYZ las siguientes:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} = T \begin{bmatrix} r_u \\ r_v \\ r_w \\ 1 \end{bmatrix} \quad \text{Ec. 2.31}$$

Y a su vez un vector $r_{x,y,z}$ rotado según T vendrá expresado por $r'_{x,y,z}$ según:

$$\begin{bmatrix} r'_x \\ r'_y \\ r'_z \\ 1 \end{bmatrix} = T \begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} \quad \text{Ec. 2.32}$$

Combinación de rotaciones y traslaciones

La principal ventaja de las matrices homogénea reside en su capacidad de representación conjunta de posición y orientación. Esta representación se realiza utilizando al mismo tiempo la matriz de rotación $R_{3 \times 3}$ y el vector de traslación $p_{3 \times 1}$ en una matriz de transformación homogénea. Es por tanto la aplicación conjunta de lo visto en los dos apartados anteriores.

La traslación y la rotación son transformaciones que se realizan en relación a un sistema de referencia. Por lo tanto, si se quiere expresar la posición y orientación de un sistema $O'UVW$, originalmente coincidente con el de referencia y que ha sido rotado y trasladado según éste, habrá que tener en cuenta si primero se ha realizado la rotación y después la traslación o viceversa, pues se trata de transformaciones espaciales no conmutativas. En la figura 2.7 se demuestra esta no conmutatividad de forma gráfica.

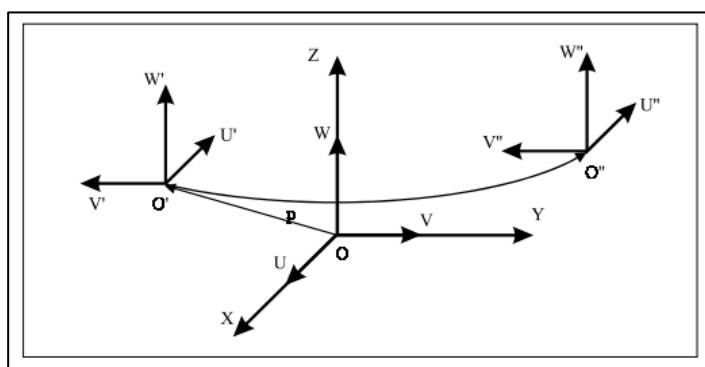


Figura 2.7. Sistemas finales según el orden de las transformaciones.

Fuente: Barrientos A., Peñín L., Balaguer C. y Aracil R. (2007). Fundamentos de robótica.

Se parte de un sistema $OUVW$ coincidente con $OXYZ$ al que se va a aplicar una traslación según un vector p_{xyz} y una rotación de 180° alrededor del eje OZ . Si primero se rota y después se traslada se obtiene un sistema final $O'U'V'W'$. En cambio, si primero se traslada y después se rota se obtiene otro sistema final $O''U''V''W''$, que representa una localización totalmente distinta a la del sistema final anterior. Se tendrá, por tanto, matrices homogéneas distintas según se realice una traslación seguida de rotación o una rotación seguida de traslación.

Rotación seguida de traslación:

Rotación de un ángulo α sobre el eje OX seguido de una traslación de vector p_{xyz} :

$$T((x, \alpha), p) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & \cos\alpha & -\sin\alpha & p_y \\ 0 & \sin\alpha & \cos\alpha & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.33}$$

Rotación de un ángulo Φ sobre el eje OY seguido de una traslación de vector p_{xyz} :

$$T((y, \phi), p) = \begin{bmatrix} \cos\phi & 0 & \text{sen}\phi & p_x \\ 0 & 1 & 0 & p_y \\ -\text{sen}\phi & 0 & \cos\phi & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.34}$$

Rotación de un ángulo θ sobre el eje OZ seguido de una traslación de vector p_{xyz} :

$$T((z, \theta), p) = \begin{bmatrix} \cos & -\text{sen} & 0 & p_x \\ \text{sen} & \cos & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.35}$$

Traslación seguida de rotación:

Traslación de vector p_{xyz} seguida de rotación de un ángulo α sobre el eje OX.

$$T(p, (x, \alpha)) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & \cos\alpha & -\text{sen}\alpha & p_y \cos\alpha - p_z \text{sen}\alpha \\ 0 & \text{sen}\alpha & \cos\alpha & p_y \text{sen}\alpha + p_z \cos\alpha \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.36}$$

Traslación de vector p_{xyz} seguida de rotación de un ángulo Φ sobre el eje OY.

$$T(p, (y, \Phi)) = \begin{bmatrix} \cos\phi & 0 & \text{sen}\phi & p_x \cos\phi + p_z \text{sen}\phi \\ 0 & 1 & 0 & p_y \\ -\text{sen}\phi & 0 & \cos\phi & p_z \cos\phi - p_x \text{sen}\phi \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.37}$$

Traslación de vector p_{xyz} seguida de rotación de un ángulo θ sobre el eje OZ

$$T(p, (z, \theta)) = \begin{bmatrix} \cos & -\text{sen} & 0 & p_x \cos - p_y \text{sen} \\ \text{sen} & \cos & 0 & p_x \text{sen} + p_y \cos \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.38}$$

Perspectiva y escalado

Las matrices homogéneas también se pueden aplicar para la realización de un escalado de las componentes de un vector. Basta utilizar una matriz T del tipo:

$$T = \begin{bmatrix} a & 0 & 0 & 0 \\ 0 & b & 0 & 0 \\ 0 & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.39}$$

Cualquier vector $r(x,y,z)$ puede ser transformado en el vector $r(ax,by,cz)$. También se puede realizar un escalado global de las tres componentes mediante la matriz:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & s \end{bmatrix} \quad \text{Ec. 2.40}$$

Cualquier vector $r(x,y,z)$ puede ser transformado en el vector $r(ax,by,cz)$.

2.1.2.3. Significado geométrico de las matrices homogéneas

Como ya se ha descrito, una matriz homogénea sirve para transformar un vector expresado en coordenadas homogéneas con respecto a un sistema $O'UVW$, a su expresión en las coordenadas del sistema de referencia $OXYZ$. También se puede utilizar para rotar y girar un vector referido a un sistema de referencia fijo, y en definitiva sirve para expresar la orientación y posición de un sistema de referencia $O'UVW$ con respecto a otro fijo $OXYZ$.

La matriz T de transformación se suele escribir de la siguiente forma:

$$T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{Ec. 2.41}$$

Donde n , o , a es una terna ortonormal que representa la orientación y P es un vector que representa la posición.

Si se considera un vector $r_{uvw} = [0,0,0,1]^T$, es decir, el origen del sistema $O'UVW$, la aplicación de la matriz T que representa la transformación (traslación + rotación) de $O'UVW$ con respecto a $OXYZ$, se obtiene r_{xyz} :

$$r_{xyz} = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} \quad \text{Ec. 2.42}$$

Que coincide con el vector columna p de T . Por tanto, este vector columna representa la posición del origen de $O'UVW$ con respecto del sistema $OXYZ$.

Si, de igual manera, se considera el vector de coordenadas homogéneas $[1,0,0,1]^T$ con respecto del sistema $O'UVW$, es decir, el vector director del eje coordenado $O'U$ del sistema $O'UVW$, y suponiendo el vector p de traslación nulo, se tendrá:

$$\begin{bmatrix} r_x \\ r_y \\ r_z \\ 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} n_x \\ n_y \\ n_z \\ 1 \end{bmatrix} \quad \text{Ec. 2.43}$$

Es decir, el vector columna n representa las coordenadas del eje O'U del sistema O'UVW con respecto del sistema OXYZ. De igual forma, si se realiza la transformación de los vectores $[0,1,0,1]^T$ y $[0,0,1,1]^T$ referidos al sistema O'UVW, se obtiene que el vector columna o representa las coordenadas del eje OY del sistema O'UVW con respecto del sistema OXYZ, y que el vector columna a representa las coordenadas del eje O'W del sistema O'UVW con respecto del sistema OXYZ.

Consecuentemente, los vectores n, o, a, definen una terna ortonormal a derechas, lo que significa que:

$$[n] = [o] = [a] = 1 \quad \text{Ec. 2.44}$$

$$n \times o = a \quad \text{Ec. 2.45}$$

Como ya se vio cuando se explicó la matriz de rotación, la submatriz de rotación [n,o,a] corresponde a una matriz ortonormal, que cumple que:

$$[n \ o \ a]^{-1} = [n \ o \ a]^T \quad \text{Ec. 2.46}$$

La matriz inversa de la matriz homogénea de transformación T es fácilmente obtenible, y corresponde a la siguiente expresión:

$$T^{-1} = \begin{bmatrix} n_x & n_y & n_z & -n^T p \\ o_x & o_y & o_z & -o^T p \\ a_x & a_y & a_z & -a^T p \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.47}$$

Si se tiene la relación $r_{xyz} = T \ r_{uvw}$ y se multiplica en ambos miembros por T^{-1} , se tiene:

$$T^{-1} r_{xyz} = r_{uvw} \quad \text{Ec. 2.48}$$

Por lo que, realizando el mismo proceso que se hizo anteriormente, se deduce que los vectores fila de la submatriz de rotación de la matriz T (vectores columna de la submatriz de rotación de T^{-1}), representan los ejes principales del sistema de coordenadas de referencia OXYZ con respecto a OUVW. Es decir, los vectores fila de la matriz [n o a] representan otra terna ortonormal a derechas.

2.1.2.4. Composición de matrices homogéneas

Anteriormente se ha mencionado que una matriz de transformación homogénea sirve, entre otras cosas, para representar el giro y la traslación realizados sobre un sistema de referencia. Esta utilidad de las matrices homogéneas cobra aún más importancia cuando se componen las matrices homogéneas para describir diversos giros y traslaciones consecutivos sobre un sistema de referencia determinado.

De esta forma, una transformación compleja podrá de componerse en la aplicación consecutiva de transformaciones simples (giros básicos y traslaciones).

Por ejemplo, una matriz que representa un giro de un ángulo α sobre el eje OX, seguido de un giro de ángulo Φ sobre el eje OY y de un giro de un ángulo θ sobre el eje OZ, puede obtenerse por la composición de las matrices básicas de rotación:

$$T = R(z, \theta) \cdot R(y, \phi) \cdot R(x, \alpha) \quad \text{Ec. 2.49}$$

$$T = \begin{bmatrix} C\phi & 0 & S\phi & 0 \\ 0 & 1 & 0 & 0 \\ -S\phi & 0 & C\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha & -S\alpha & 0 \\ 0 & S\alpha & C\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.50}$$

$$T = \begin{bmatrix} C\phi C\theta & -S\theta C\alpha C\theta S\phi S\alpha & S\theta S\alpha + C\theta S\phi C\alpha & 0 \\ S\theta C\phi & C\theta C\alpha + S\theta S\phi S\alpha & -C\theta S\alpha + S\theta S\phi C\alpha & 0 \\ -S\phi & C\phi S\alpha & C\alpha C\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.51}$$

Si se invierte el orden de aplicación, el resultado es distinto.

$$T = R(x, \alpha) \cdot R(y, \phi) \cdot R(z, \theta) \quad \text{Ec. 2.52}$$

$$T = \begin{bmatrix} C\phi C\theta & -C\phi S\theta & S\phi & 0 \\ S\alpha S\phi C\theta + C\alpha S\theta & -S\alpha S\phi S\theta + C\alpha C\theta & -S\alpha C\phi & 0 \\ -C\alpha S\phi C\theta & C\alpha S\phi S\theta + S\alpha C\theta & C\alpha C\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.53}$$

2.1.2.5. Gráficos de transformación

Es frecuente encontrar situaciones en las que la localización espacial de un objeto o de su sistema de referencia asociado, pueda realizarse a través de la composición de diversas transformaciones distintas. En la figura 2.8 se tiene un manipulador cuya base está referida al sistema del mundo OXYZ mediante la transformación ${}^M T_R$. A su vez, para pasar de la base del manipulador a su extremo se utiliza la transformación ${}^R T_E$.

El extremo de la herramienta está referido con respecto al extremo del manipulador por la transformación ${}^E T_H$. A su vez, su objeto está referido con respecto al sistema OXYZ mediante la transformación ${}^M T_O$, y por último, el extremo de la herramienta está referido con respecto al objeto a través de la transformación ${}^O T_H$.

Se observa que el final de la herramienta puede ser referido con respecto al sistema OXYZ de dos maneras distintas: a través del manipulador y a través del objeto. De tal manera que se puede escribir:

$${}^M T_R {}^R T_E {}^E T_H = {}^M T_O {}^O T_H \quad \text{Ec. 2.54}$$

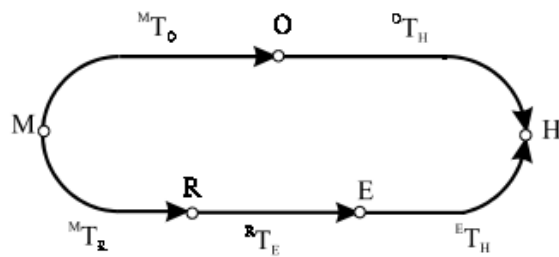


Figura 2.8. Gráfico de transformación

Fuente: Barrientos A., Peñin L., Balaguer C. y Aracil R. (2007). Fundamentos de robótica.

Esta relación se puede representar mediante un gráfico de transformación, de tal manera que si se quiere obtener la relación entre el objeto y la herramienta bastará multiplicar ambos miembros de la ecuación anterior por ${}^M T_O^{-1}$ obteniéndose:

$$({}^M T_O)^{-1} {}^M T_R {}^R T_E {}^E T_H = {}^O T_H \quad \text{Ec. 2.55}$$

Cualquier otra relación puede ser obtenida fácilmente a partir del gráfico. Para ello se irá desde el objeto inicial al final multiplicando las matrices de transformación correspondiente a los arcos del gráfico, y considerando que de recorrerse éstos en el sentido inverso a las flechas se deberá utilizar una matriz inversa. Así la relación entre la base del robot y el objeto vendrá dada por:

$${}^R T_O = {}^R T_E {}^E T_H ({}^O T_H)^{-1} \quad \text{Ec. 2.56}$$

O bien por:

$${}^R T_O = ({}^M T_R)^{-1} {}^M T_O \quad \text{Ec. 2.57}$$

2.2 Comparación de métodos de localización espacial

La comparación se realiza fundamentalmente en razón a su capacidad para la realización de cuatro cuestiones básicas de toda transformación:

- Capacidad de representación conjunta de posición y orientación.
- Representar la posición y orientación de un sistema rotado y trasladado $O'UVW$ con respecto a un sistema fijo de referencia $OXYZ$. Que es lo mismo que representar una rotación y traslación realizada sobre un sistema de referencia.
- Transformar un vector expresado en coordenadas con respecto a un sistema $O'UVW$, a su expresión en coordenadas del sistema de referencia $OXYZ$.
- Rotar y trasladar un vector con respecto a un sistema de referencia fijo $OXYZ$.

Las ventajas de las matrices de transformación homogénea sobre otros métodos residen en su capacidad de representación conjunta de posición y orientación y en la comodidad con la que se puede realizar la composición de transformaciones. Para ello bastará únicamente multiplicar, en el orden adecuado, las matrices de transformación correspondientes. Es posible, además, la aplicación de una transformación sobre un vector referido a un sistema fijo únicamente multiplicando la matriz de transformación correspondiente por el vector. Su principal inconveniente es su alto nivel de redundancia (necesita definir 12 componentes para sólo 6 grados de libertad). Esto dificulta su implementación en computadoras.

2.3 Cinemática de un robot

La cinemática del robot estudia el movimiento del mismo con respecto a un sistema de referencia y se interesa por la descripción analítica del movimiento espacial del robot como una función del tiempo, y en particular por las relaciones entre la posición y la orientación del extremo final del robot con los valores que toman sus coordenadas en las articulaciones. Hasta aquí, el método más utilizado es el de Denavit y Hartenberg, quienes propusieron una forma de describir y representar la geometría espacial de los elementos de una “cadena cinemática” de un robot, con respecto a un sistema de referencia fijo, mediante una matriz de transformación homogénea. Para encontrar las relaciones entre las velocidades del movimiento de las articulaciones y las del extremo se utiliza el “modelo diferencial” expresado mediante una matriz Jacobiana.

Existen dos problemas fundamentales a resolver en la cinemática del robot (figura 2.9), éstos son:

- Problema cinemático directo: consiste en determinar cuál es la posición y orientación del extremo final del robot, con respecto a un sistema de coordenadas que se toma como referencia, si se conocen los valores de las articulaciones y los parámetros geométricos de los elementos del robot
- Problema cinemático inverso: permite resolver la configuración geométrica que debe adoptar el robot para una posición y orientación del extremo conocidas.

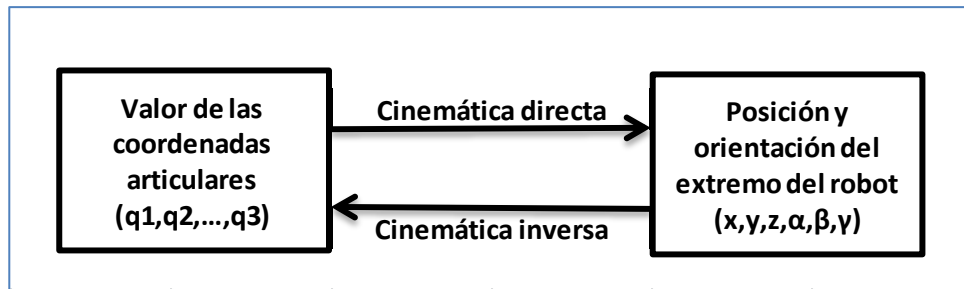


Figura 2.9. Diagrama de relación entre cinemática directa e inversa.

Fuente: Barrientos A., Peñín L., Balaguer C. y Aracil R. (2007). Fundamentos de robótica.

2.3.1 Problema cinemático directo

Un robot se puede considerar como una cadena cinemática formada por objetos rígidos o eslabones unidos entre sí, mediante articulaciones. En esta situación se puede establecer un sistema de referencia fijo en la base del robot y describir la localización de cada uno de los eslabones con respecto a dicho sistema de referencia.

El problema cinemático directo se reduce a encontrar una matriz de transformación homogénea T que relacione la posición y orientación del extremo del robot respecto del sistema de referencia fijo, situado en la base del mismo.

En general, un robot de n grados de libertad está formado por n eslabones unidos por n articulaciones, de forma que cada par “articulación-eslabón” constituye un grado de libertad. A cada eslabón se le puede asociar un sistema de referencia solidario a él. Mediante las matrices de transformación homogéneas es posible representar las rotaciones y traslaciones relativas entre los distintos eslabones que componen el robot. Normalmente, la matriz de transformación homogénea se suele denominar matriz ${}^{i-1}A_i$ que representa la posición y orientación del sistema de referencia solidario del eslabón i con respecto al sistema de referencia solidario al eslabón $i-1$. Para facilidad de notación, se define la matriz 0A_k como la resultante del producto de las matrices ${}^{i-1}A_i$ desde $i=1$ a $i=k$. Si se consideran todos los grados de libertad de un robot (sean éstos p) entonces se tiene que la matriz homogénea T general se obtiene como:

$$T = {}^0A_1 * {}^1A_2 * {}^2A_3 * \dots * {}^{i-1}A_i * \dots * {}^{p-1}A_p \quad \text{Ec. 2.58}$$

2.3.1.1 Resolución del problema cinemático directo mediante matrices de transformación homogénea

Una forma simple para describir la relación que existe entre dos elementos contiguos fue propuesta en 1955 por Denavit y Hartenberg (D-H). Según la representación D-H, si se escoge adecuadamente los sistemas de coordenadas asociados a cada eslabón, entonces es posible pasar de un eslabón al siguiente mediante 4 transformaciones básicas que dependen exclusivamente de las características geométricas del eslabón.

Estas transformaciones básicas consisten en una sucesión de rotaciones y traslaciones que permiten relacionar el sistema de referencia del elemento i , con el sistema del elemento $i-1$. Las transformaciones en cuestión son las siguientes y deben hacerse en este orden:

1. Rotación alrededor del eje Z_{i-1} un ángulo θ_i
2. Traslación a lo largo del eje Z_{i-1} una distancia d_i ; vector $d_i(0,0,d_i)$
3. Traslación a lo largo del eje X_i una distancia a_i ; vector $a_i(0,0,a_i)$
4. rotación alrededor del eje X_i un ángulo α

Las transformaciones antes indicadas se pueden expresar mediante las siguientes matrices homogéneas:

$${}^{i-1}A_1 = T(Z, \theta_i) * T(0, 0, d_i) * T(a_i, 0, 0) * T(X, \alpha_i) \quad \text{Ec. 2.59}$$

$${}^{i-1}A_1 = T(Z, \theta_i) * T(d_i) * T(a_i) * T(X, \alpha_i) \quad \text{Ec. 2.60}$$

$${}^{i-1}A_1 = \begin{bmatrix} C & -S & 0 & 0 \\ S & C & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & a_i \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha & -S\alpha & 0 \\ 0 & S\alpha & C\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.61}$$

$${}^{i-1}A_1 = \begin{bmatrix} C_{\theta_i} & -C_{\alpha_i}S_{\theta_i} & S_{\alpha_i}S_{\theta_i} & a_iC_{\theta_i} \\ S_{\theta_i} & C_{\alpha_i}C_{\theta_i} & -S_{\alpha_i}C_{\theta_i} & a_iS_{\theta_i} \\ 0 & S_{\alpha_i} & C_{\alpha_i} & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 2.62}$$

Los valores θ_i , a_i , d_i , α_i se denominan parámetros D-H del eslabón i . De este modo, basta con identificar tales parámetros para obtener las matrices A y relacionar así todos y cada uno de los eslabones del robot.

La matriz ${}^{i-1}A_i$ relaciona los sistema de referencia S_i y S_{i-1} y es necesario que los sistemas de referencia se hayan escogido de acuerdo a ciertas normas. Estas, junto con los parámetros D-H, conforman el siguiente algoritmo para la resolución del problema cinemático.

2.3.1.2 Algoritmo de Denavit – Hartenberg

D-H 1: numerar los eslabones comenzando con 1 (primer eslabón móvil de la cadena) y finalizando con n (último eslabón móvil). Se numerará como eslabón 0 a la base fija del robot.

D-H 2: numerar cada articulación comenzando por 1 (la correspondiente al primer grado de libertad) y finalizando en n .

D-H 3: localizar el eje de cada articulación. Si ésta es rotativa, el eje será su propio eje de giro. Si es prismática, será el eje a lo largo del cual se produce el desplazamiento.

D-H 4: para i de 0 a $n-1$, situar el eje Z_i sobre el eje de la articulación $i+1$.

D-H 5: situar el origen del sistema de la base (S_0) en cualquier punto del eje Z_0 . Los ejes X_0 e Y_0 se situarán de modo que formen un “sistema dextrógiro” (giro a la derecha) con el eje Z_0 .

D-H 6: para i de 1 a $n-1$, situar el sistema S_i (solidario al eslabón i) en la intersección del eje Z_i con la línea normal común a los ejes Z_{i-1} y Z_i . Si ambos ejes se cortasen, se situaría el sistema S_i en el punto de corte. Si fuesen paralelos, el sistema S_i se situaría en la articulación $i+1$.

D-H 7: situar el eje X_i en la línea normal común a los ejes Z_{i-1} y Z_i .

D-H 8: situar Y_i de modo que forme un sistema dextrógiro con los ejes X_i y Z_i .

D-H 9: situar el sistema S_n en el extremo del robot de modo que Z_n coincida con la dirección de Z_{n-1} y que X_n sea normal a Z_{n-1} y Z_n .

D-H 10: obtener θ_i como el ángulo que hay que girar en torno al eje Z_{i-1} para que los ejes X_{i-1} y X_i queden paralelos.

D-H 11: obtener d_i como la distancia, medida a lo largo del eje Z_{i-1} , que habría que desplazar el sistema S_{i-1} para que los ejes X_i y X_{i-1} quedasen alineados.

D-H 12: obtener a_i como la distancia medida a lo largo del eje X_i (que ahora coincidiría con el eje X_{i-1}) que habría que desplazar el nuevo sistema S_{i-1} para que su origen coincidiese con el sistema S_i .

D-H 13: obtener α_i como el ángulo que habría que girar entorno al eje X_i (que ahora coincidiría con el eje X_{i-1}), para que el nuevo sistema S_{i-1} coincidiese totalmente con el sistema S_i .

D-H 14: obtener las matrices de transformación ${}^{i-1}A_i$.

D-H 15: obtener la matriz de transformación que relaciona el sistema de la base con el del extremo del robot, esto es, $T = {}^0A_1 * {}^1A_2 * {}^2A_3 * \dots * {}^{n-1}A_n$.

D-H 16: la matriz T define la orientación y posición del extremo referido a la base en función de las n coordenadas articular

Los cuatro parámetros de D-H dependen únicamente de las características geométricas de cada eslabón y de las articulaciones que las unen con la anterior y siguiente (figura 2.10).

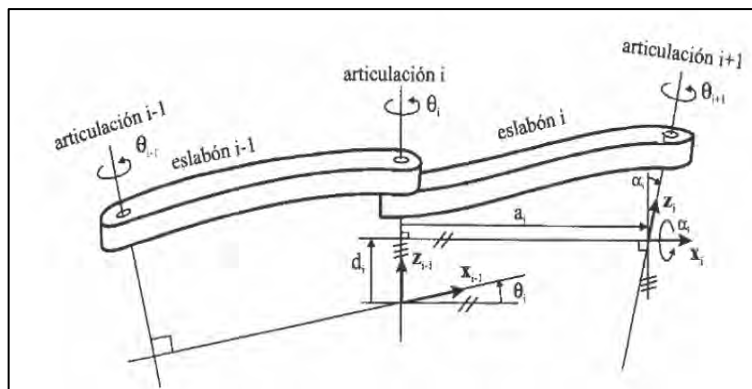


Figura 2.10. Parámetros D-H para un eslabón giratorio.

Fuente: Barrientos A., Peñin L., Balaguer C. y Aracil R. (2007). Fundamentos de robótica.

- θ_i es el ángulo que forman los ejes X_{i-1} y X_i medido en un plano perpendicular al eje Z_{i-1} , utilizando la regla de la mano derecha. Se trata de un parámetro variable en articulaciones giratorias.
- d_i es la distancia a lo largo del eje Z_{i-1} desde el origen del sistema de coordenadas $(i-1)$ -ésimo hasta la intersección del eje Z_{i-1} con el eje X_i . Se trata de un parámetro variable en articulaciones prismáticas.
- a_i es la distancia a lo largo del eje X_i que va desde la intersección del eje Z_{i-1} con el eje X_i hasta el origen del sistema i -ésimo, en el caso de articulaciones giratorias. En el caso de articulaciones prismáticas, se calcula como la distancia más corta entre los ejes Z_{i-1} y Z_i .
- α_i es el ángulo de separación del eje Z_{i-1} y el eje Z_i , medido en un plano perpendicular al eje X_i , utilizando la regla de la mano derecha.

Una vez obtenidos los parámetros D-H, el cálculo de las relaciones entre los eslabones consecutivos del robot es inmediato, porque vienen dadas por las matrices A. Las relaciones entre eslabones no consecutivos viene dadas por las matrices T, que se obtiene como producto de un conjunto de matrices A.

2.3.1.3 Aplicación del algoritmo de D-H en el brazo robot

La figura 2.11 esquematiza el brazo digitalizador construido para el respectivo análisis cinemático, donde se ha seguido cada uno de los pasos para establecer los parámetros Denavit-Hartenberg.

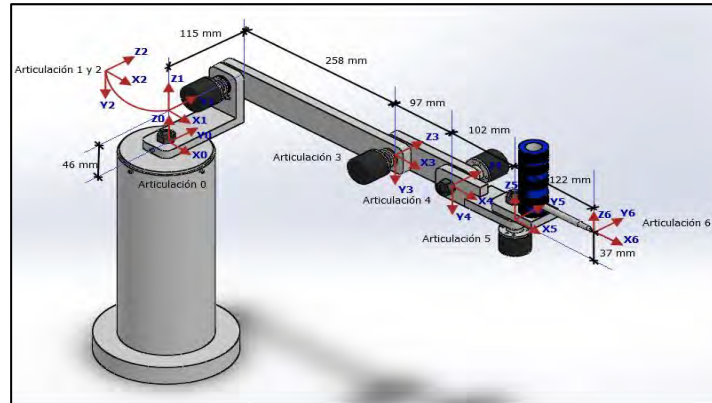


Figura 2.11. Formulación Denavit – Hartenberg. Posición inicial del brazo robot

Fuente: elaboración propia.

Paso 1: para desarrollar la formulación Denavit-Hartenberg se debe identificar cada una de las articulaciones (figura 2.12) que brindan cada uno de los giros del brazo robot y están representadas por " q_i " ($q_1, \dots, q_5$). Son cinco articulaciones que proporcionan los cinco grados de libertad al brazo robot. Luego, se colocan los ejes Z_1 hasta Z_5 , que representan al eje de giro de cada una de las articulaciones.

Paso 2: se selecciona el sistema de referencia fijo (figura 2.12) ubicada en la base del robot que coincide con la ubicación de la primera articulación y cuyo origen será llamado " O_0 " a lo largo de Z_0 . Se coloca X_0 y Y_0 de forma que se genere un sistema dextrógiro o sistema que cumpla con la regla de la mano derecha. Se advierte que este sistema deberá generar que algún parámetro Denavit-Hartenberg sea igual a cero: θ_i (rad), d_i (mm), a_i (mm), α_i (rad).

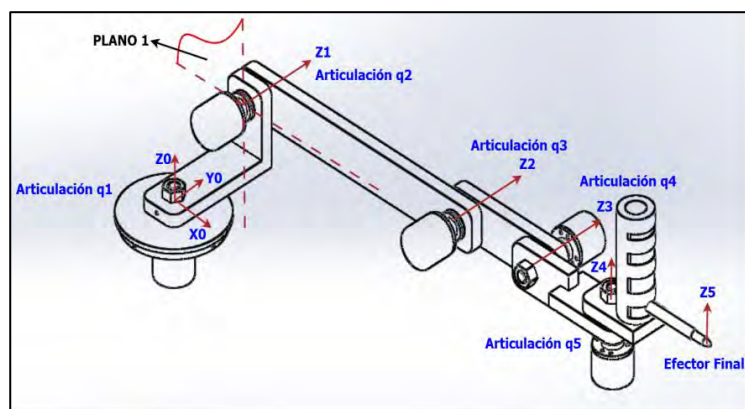


Figura 2.12. Identificación de las articulaciones y selección del sistema de referencia X_0, Y_0, Z_0 .

Fuente: elaboración propia.

Paso 3: El sistema de referencia o base del robot genera un parámetro D-H igual a cero, por lo tanto, se verifican las dos condiciones importantes para aplicar este método, que X_i sea perpendicular a Z_{i-1} y que X_i y Z_{i-1} se intersecten. Los ejes X_1 y Y_1 deberán ubicarse en el plano, por lo tanto no habrá intersección entre X_1 y Z_0 , y así no habrá tampoco ningún parámetro Denavit-Hartenberg igual a cero.

Una forma de solucionar esto es generando una articulación ficticia que permita cumplir con las condiciones anteriores (figura 2.13); de esta manera se ha creado el sistema de referencia X_1, Y_1 y Z_1 , y que deberá tener la misma configuración que el sistema de referencia anterior dado que el parámetro θ_i (rad) de esta articulación, coincidirá con el mismo parámetro de la articulación anterior, representando el mismo giro. Ahora existe intersección entre X_1 y Z_0 y el parámetro Denavit-Hartenberg " a_i (mm)" es igual a cero. Cabe resaltar que el parámetro θ_i (rad) de esta articulación, coincidirá con el mismo parámetro de la articulación anterior dado que representan el mismo giro.

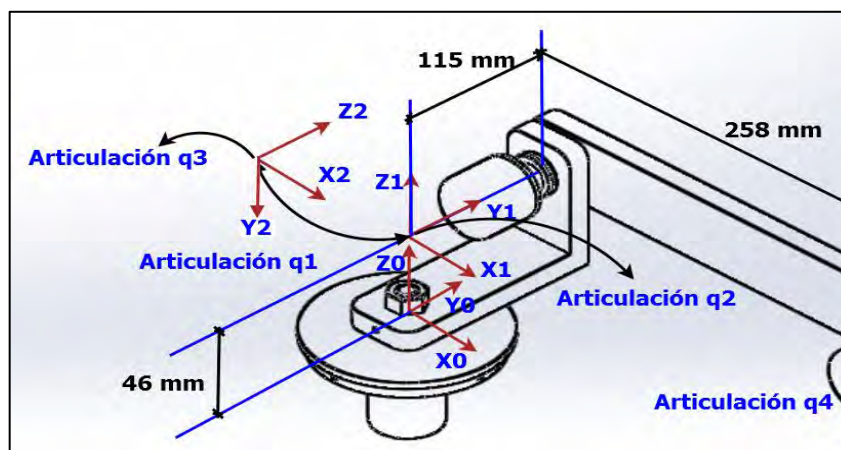


Figura 2.13. Generación de una articulación ficticia X_1, Y_1, Z_1 .
Fuente: elaboración propia.

La siguiente articulación " q_3 " deberá coincidir con la articulación anterior para que siga existiendo intersección entre los ejes referidos y para que sigan existiendo a favor, parámetros Denavit-Hartenberg iguales a cero. Vemos que ahora la articulación " q_3 " representa el giro que realiza verdaderamente la articulación número 2.

Para las siguientes articulaciones ya no existirá ningún problema, así que se seguirá considerando las condiciones del paso 3. El análisis culminado del paso 3 es mostrado en la figura 2.14.

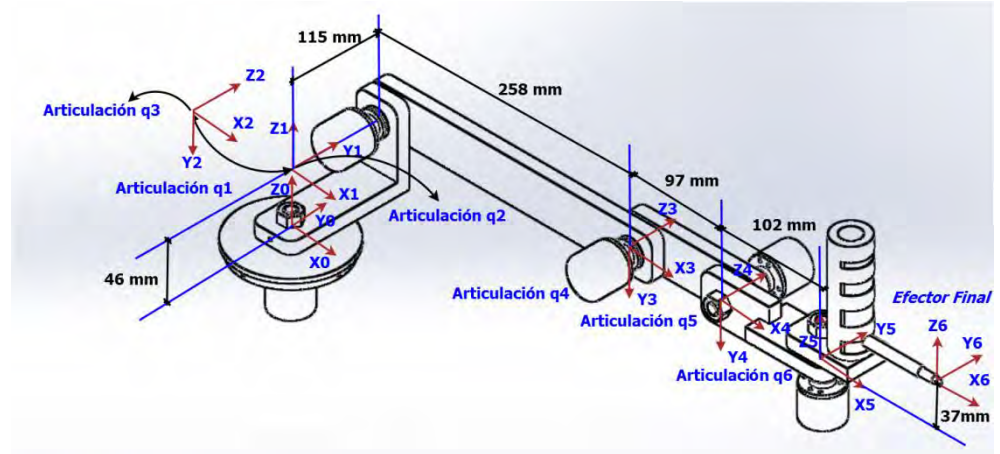


Figura 2.14. Sistema de referencia para el efector final.

Fuente: elaboración propia.

Paso 4: En la figura 2.14 se ha establecido un sistema de referencia para el efector final, el cual se ha definido mediante los ejes X_6, Y_6, Z_6 y que generalmente siguen la configuración de la referencia anterior.

Paso 5.- Completar la tabla de Denavit-Hartenberg (tabla 2.1) con los cuatro parámetros para cada articulación en cada fila, para ángulos iniciales iguales a $\theta_1 = \theta_2 = \theta_3 = \theta_4 = \theta_5 = \theta_6 = 0$.

Tabla 2.1 Valores de ingreso para prueba 1.

PRUEBA 1	θ_i (rad)	d_i (mm)	a_i (mm)	α_i (rad)
Articulación 1	0	46	0	0
Articulación 2	0	0	0	$-\pi/2$
Articulación 3	0	115	258	0
Articulación 4	0	0	97	0
Articulación 5	0	0	102	$\pi/2$
Articulación 6	0	37	122	0

Fuente: elaboración propia.

Paso 6: calcular ${}^{i-1}_iH(q_i)$ para posteriormente calcular ${}^0_nH(q)$. Para ello se reemplazan los valores de los parámetros para cada articulación anterior de la tabla 2.1 en la matriz de cambio de base (Ec. 2.62) cuya forma extendida se muestra de la siguiente forma.

$${}^{i-1}_iH = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) * \cos(\alpha_i) & \sin(\theta_i) * \sin(\alpha_i) & a_i * \cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) * \cos(\alpha_i) & -\cos(\theta_i) * \sin(\alpha_i) & a_i * \sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matriz de cambio de base para la articulación 1:

$${}^0_1H = \begin{pmatrix} \cos(0) & -\sin(0) * \cos(0) & \sin(0) * \sin(0) & 0 * \cos(0) \\ \sin(0) & \cos(0) * \cos(0) & -\cos(0) * \sin(0) & 0 * \sin(0) \\ 0 & \sin(0) & \cos(0) & 46 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^0_1H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 46 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matriz de cambio de base para la articulación 2:

$${}^1_2H = \begin{pmatrix} \cos(0) & -\sin(0) * \cos(-\pi/2) & \sin(0) * \sin(-\pi/2) & 0 * \cos(0) \\ \sin(0) & \cos(0) * \cos(-\pi/2) & -\cos(0) * \sin(-\pi/2) & 0 * \sin(0) \\ 0 & \sin(-\pi/2) & \cos(-\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^1_2H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matriz de cambio de base para la articulación 3:

$${}^2_3H = \begin{pmatrix} \cos(0) & -\sin(0) * \cos(0) & \sin(0) * \sin(0) & 258 * \cos(0) \\ \sin(0) & \cos(0) * \cos(0) & -\cos(0) * \sin(0) & 258 * \sin(0) \\ 0 & \sin(0) & \cos(0) & 115 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^2_3H = \begin{pmatrix} 1 & 0 & 0 & 258 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 115 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matriz de cambio de base para la articulación 4:

$${}^3_4H = \begin{pmatrix} \cos(\pi/2) & -\sin(\pi/2) * \cos(0) & \sin(\pi/2) * \sin(0) & 97 * \cos(0) \\ \sin(\pi/2) & \cos(\pi/2) * \cos(0) & -\cos(\pi/2) * \sin(0) & 97 * \sin(0) \\ 0 & \sin(0) & \cos(0) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^3_4H = \begin{pmatrix} 1 & 0 & 0 & 97 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matriz de cambio de base para la articulación 5:

$${}^4_5H = \begin{pmatrix} \cos(0) & -\sin(0) * \cos(\pi/2) & \sin(0) * \sin(\pi/2) & 102 * \cos(0) \\ \sin(0) & \cos(0) * \cos(\pi/2) & -\cos(0) * \sin(\pi/2) & 102 * \sin(0) \\ 0 & \sin(\pi/2) & \cos(\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^4_5H = \begin{pmatrix} 1 & 0 & 0 & 102 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Matriz de cambio de base para la articulación 6:

$${}^5_6H = \begin{pmatrix} \cos(0) & -\sin(0) * \cos(0) & \sin(0) * \sin(0) & 122 * \cos(0) \\ \sin(0) & \cos(0) * \cos(0) & -\cos(0) * \sin(0) & 122 * \sin(0) \\ 0 & \sin(0) & \cos(0) & 37 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$${}^5_6H = \begin{pmatrix} 1 & 0 & 0 & 122 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 37 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Paso 7: calculando ahora

$${}^0_6H = {}^0_1H * {}^1_2H * {}^2_3H * {}^3_4H * {}^4_5H * {}^5_6H$$

$${}^0_6H = \begin{pmatrix} 1 & 0 & 0 & 579 \\ 0 & 1 & 0 & 115 \\ 0 & 0 & 1 & 83 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Calculando, las coordenadas del punto "P" que representa el extremo del brazo digitalizador son:

$$\begin{aligned} p_x &= 579 \text{ mm} \\ p_y &= 115 \text{ mm} \\ p_z &= 83 \text{ mm} \end{aligned}$$

2.3.2 Cinemática inversa

La cinemática inversa es la técnica que permite determinar el movimiento de una cadena de articulaciones para lograr que un actuador final se ubique en una posición concreta.

El cálculo de la cinemática inversa es un problema complejo que consiste en la resolución de una serie de ecuaciones cuya solución normalmente no es única.

El objetivo de la cinemática inversa es encontrar los valores que deben tomar las coordenadas articulares del robot para que su extremo se posicione y oriente según una determinada localización espacial. Depende de la configuración del robot y existen soluciones múltiples.

No se desarrollará la problemática inversa porque no es de aplicación para este trabajo de investigación.

Capítulo 3

Diseño y ensamble del brazo robot

3.1 Confrontación del problema

3.1.1 Formulación del problema

La idea de diseñar un brazo robot que permita escanear curvaturas, nace de la necesidad de reproducir piezas con curvaturas complicadas de medición, cuando éstas han sufrido algún tipo de daño que motive un mantenimiento correctivo en los sistemas de producción de cualquier industria. En algunos casos, cuando la maquinaria es antigua, es difícil encontrar piezas similares o resulta demasiado costosa su compra directa con un proveedor.

También, según los trabajadores del Laboratorio de Ingeniería Mecánico-Eléctrica de la Universidad de Piura, se ha podido determinar que existe dificultad para:

- Reproducción del perfil de levas.
- Curvatura del rodete de las bombas centrífugas.
- Curvatura de los álabes de la turbina Pelton.
- Curvatura de las paletas de los rodetes de las turbinas Francis y Kaplan.
- Curvatura o diámetro de orificios que se encuentran al interior de piezas como culata de un motor de combustión interna y que resulta difícil obtener su dimensión.

Otra de las motivaciones resultó de observar la importancia que hoy en día está adquiriendo la robótica en la industria y sobre todo, los fundamentos y conocimientos matemáticos con los cuales se desarrollan y que resultan atractivos para los alumnos de Ciencias de la ingeniería, porque en un brazo robótico se podrán poner en práctica conceptos físicos, matemáticos y de software vistos, aprendidos y utilizados a lo largo de la carrera universitaria.

3.1.2 Análisis FODA

Para llegar a la concepción de este proyecto, se consideró importante realizar un estudio que permita tomar decisiones acertadas al momento de trazar una visión futura de este proyecto, por tal motivo, se realizó un análisis FODA, por su gran utilidad y sencillez y que comprende un estudio personal de la situación del

entorno interno (Debilidades y Fortalezas) y externo (Amenazas y Oportunidades), maximizando el potencial de las fuerzas y oportunidades minimizando el impacto de las debilidades y amenazas. La figura 3.1 esquematiza el FODA.

FACTORES CLAVE		
Contexto	Factores positivos	Factores Negativos
Interno	Fortalezas	Debilidades
Externo	Oportunidades	Amenazas

Figura 3.1. Factores clave del FODA

Fuente: Díaz L. (2005).Análisis y Planeamiento..

La parte interna se relaciona con los aspectos sobre los cuales el planificador tiene algún grado de control. Se trata de la identificación de las fortalezas y debilidades del proyecto.

- Fortalezas: Capacidades y recursos internos que puedan ser aplicados en la realización de un proyecto, incluye competencias y talentos. Se tiene un control sobre ellas.
- Debilidades: Características o capacidades internas que no contribuyen y alejan de la realización de una meta o proyecto, provocando situaciones desfavorables. Al igual que las fortalezas, también se tiene control sobre ellas.

La parte externa revela las oportunidades que ofrecen el mercado y las amenazas claves que debe enfrentar el proyecto en su entorno. Dado que se tiene poco o ningún y la habilidades del jefe del proyecto al aprovechar las oportunidades y para minimizar o anular esas amenazas.

- Oportunidades: Factores externos que se presentan para poder ser aprovechados para la realización de un proyecto, no se tiene control directo sobre ellos pero se pueden obtener ventajas que aporten al proyecto.
- Amenazas: Factores externos no controlables que pueden afectar desfavorablemente el desarrollo del proyecto.

La tabla 3.1 presenta el análisis FODA aplicado al proyecto, cuya utilidad radica en diseñar las estrategias para utilizar las fortalezas en forma tal que se puedan aprovechar las oportunidades, enfrentar las amenazas y superar las debilidades.

Tabla 3.1. Análisis FODA aplicado al prototipo de brazo robótico

FORTALEZAS	DEBILIDADES
• Conocimientos de programación y facilidad para programar de algoritmos y métodos numéricos en Arduino y Matlab. • Experiencia en trabajos y proyectos con comunicación serial Arduino – Matlab • Experiencia en proyectos de diseño mecánico de rodamientos, cadenas, cojinetes, etc. • Facilidad para aprender temas nuevos • Perseverancia, responsabilidad y capacidad de innovación y creación	• Escaso conocimiento de programación en otros lenguajes o sistemas de adquisición de datos • Falta de conocimientos en programas de ajuste de curvas • Escaso conocimientos en robótica y matrices de transformación. • En la bibliografía no se encontró ningún diseño de brazo robot para lectura de piezas con curvatura.
OPORTUNIDADES	AMENAZAS
• Disponibilidad del asesor de tesis para el desarrollo del proyecto propuesto. • Disponibilidad del Matlab y Arduino. • Acceso a información y dispositivos de lectura de datos de encoders incrementales en Arduino. • Accesibilidad a los equipos, instrumentos, dispositivos y materiales para la construcción del prototipo • Laboratorio de Mecánica de la Universidad de Piura con personal idóneo para acondicionado y fabricación de piezas. • Laboratorios de electrónica y sistemas automáticos en la UDEP. • Acceso a bibliografía especializada.	• Falta de recursos económicos • Avance y cambio tecnológico constante que origine una desactualización del brazo robot escáner. • Escaso tiempo disponible para la realización del proyecto.

Fuente: elaboración propia

3.2 Selección de los conceptos de diseño y construcción

3.2.1 Fundamento de los cinco grados de libertad

Uno de los principales parámetros para la construcción de un brazo robot son los grados de libertad, porque éstos determinarán su accesibilidad y los movimientos del efector final. En algunos casos el número de articulaciones coincide con el número de grados de libertad; aunque, dependiendo del tipo de articulación estos parámetros pueden diferir.

Para definir los grados de libertad, se debe definir las funciones que ha de realizar el brazo robótico. En algunas aplicaciones en las que se busca orientar y rotar el brazo respecto a un solo eje sólo bastan 3 grados de libertad, en otras aplicaciones de mayor dificultad o donde se requiera un mayor control del efector final, se necesitan hasta 6 grados de libertad, resaltando que al aumentar los grados de libertad se aumenta el costo del brazo robot.

Para la función escaneo, objetivo de este proyecto de tesis, se decidió utilizar un brazo robótico de cinco grados de libertad, porque se asemeja a la anatomía

humana tal como se puede apreciar en la figura 3.1 y proporcionará los movimientos necesarios para realizar el escaneo de curvaturas complicadas.

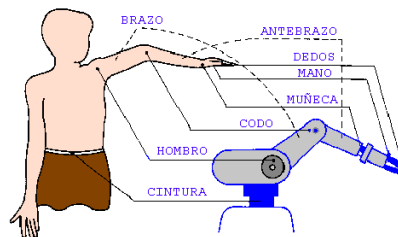


Figura 3.2 Similitud del brazo robot con la anatomía humana.

Fuente: Martínez G., Jáquez S., Rivera J., Sandoval R. (2008). Diseño y construcción de un brazo robótico de 5 GDL

3.2.1.1 Partes principales del brazo robot

La construcción de las partes del brazo robot fueron desarrolladas en el departamento de Ingeniería Mecánica Eléctrica de la Universidad de Piura, haciendo uso de sus laboratorios, sus herramientas y máquinas como: fresadora, torno, cepilladora, taladro, moladora, lijas, etc., y con el apoyo de su personal profesional y técnico. El detalle de cada una de las piezas utilizadas para la construcción del brazo robot se muestra en el anexo A.

En la figura 3.3, diseñada en Solidworks, se nombra cada una de las partes que constituyen la estructura básica del brazo robótico.

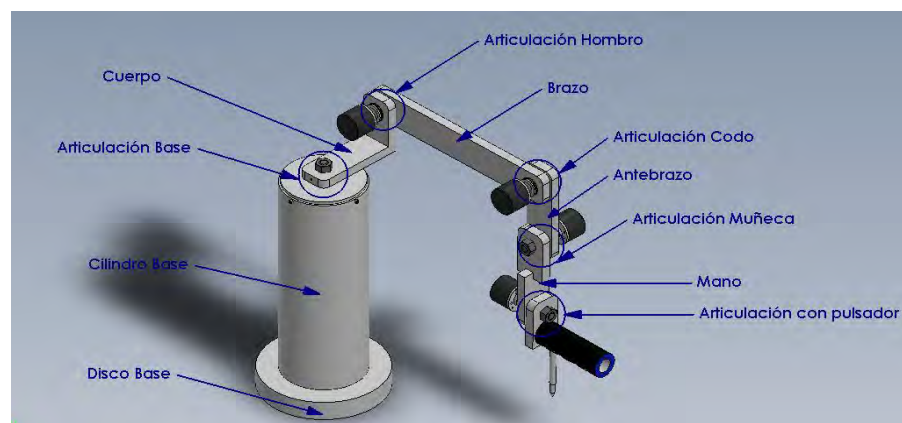


Figura 3.3. Prototipo de brazo robótico para escaneo de curvas.

Fuente; elaboración propia.

3.2.2 Análisis estático del brazo robot

El escaneo de curvas se debe realizar muy lentamente para una mejor captura y procesamiento de información; por ello, se puede concluir que bastará realizar un análisis estático de la estructura para determinar el peso mínimo de la base para evitar el fenómeno de volcadura.

El análisis estático desprecia el peso de las articulaciones y se centra en las dos posiciones extremas, la mínima y máxima distancia de escaneo, tal como se muestra en la figura 3.4.

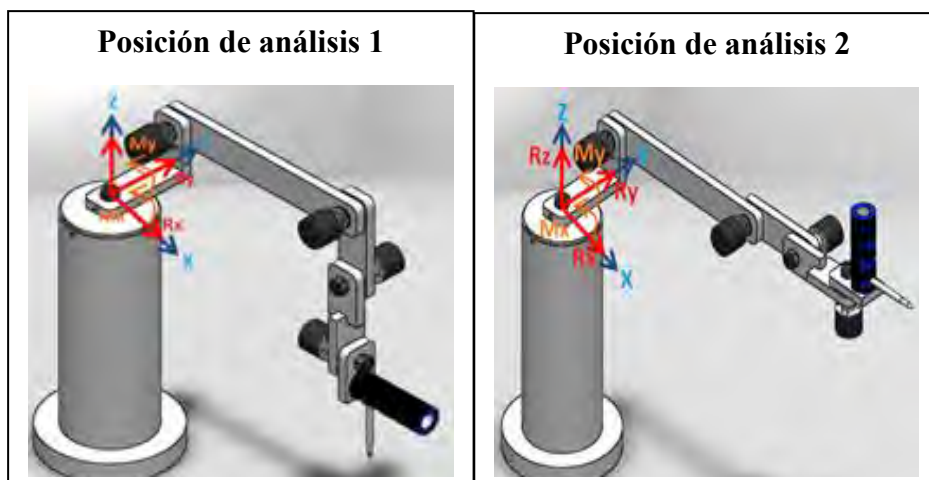


Figura 3.4. Posiciones de análisis estático del brazo robot.

Fuente: elaboración propia.

Se pesan los eslabones y se calculan sus centros de gravedad.. El detalle de los cálculos del centro de gravedad se presenta en el anexo B, pero el cuadro resumen se presenta en la tabla 3.2.

Tabla 3.2. Cálculo del centro de gravedad del prototipo de brazo robot.

Posición de análisis 1 :								
Item	Componente	Masa (Kg)	Peso (N)	Centro de Gravedad respecto al eje principal			M_x	M_y
				x (mm)	y (mm)	z (mm)		
1	Cuerpo	0,125	1,23	-0,01	57,09	5,9	0,0701	0
2	Brazo	0,202	1,98	115,84	117,35	46,02	0,2327	0,2297
3	Antebrazo	0,091	0,89	258,05	135,36	9,51	0,1209	0,2306
4	mano	0,092	0,90	258,03	119,26	-91,82	0,1077	0,2331
5	componente final	0,068	0,67	284,81	117,88	-161,82	0,0787	0,1901
6	pulsador	0,05	0,49	298,2	117,9	-218,2	0,0579	0,1464
Sumatoria de Momentos							0,66804	1,03
Posición de análisis 2 :								
Item	Componente	Peso (Kg)	Peso (N)	Coordenadas del Centro de Gravedad			M_x	M_y
				x (mm)	y (mm)	z (mm)		
1	Cuerpo	0,125	1,23	-0,01	57,09	5,9	0,0701	0
2	Brazo	0,202	1,98	115,84	117,35	46,02	0,2327	0,2297
3	Antebrazo	0,091	0,89	249,49	135,36	46,05	0,1209	0,2229
4	mano	0,092	0,90	395,82	119,26	46,03	0,1077	0,3575
5	componente final	0,068	0,67	465,82	117,88	63,44	0,0787	0,3110
6	pulsador	0,05	0,49	526,2	117,9	86,2	0,0579	0,2583
Sumatoria de Momentos							0,66804	1,38

Fuente: elaboración propia.

La sumatoria de momentos en el eje X y en el eje Y es la misma para cualquier posición; por lo tanto, se necesita un solo análisis estático.

Aplicando las condiciones de equilibrio, se tiene que:

$$\sum F_X = 0 \quad \text{Ec. 3.1}$$

$$\sum F_Y = 0 \quad \text{Ec. 3.2}$$

$$\sum F_Z = 0 \quad \text{Ec. 3.3}$$

$$R_X = 0 \quad \text{Ec. 3.4}$$

$$R_Y = 0 \quad \text{Ec. 3.5}$$

Haciendo sumatoria de fuerzas en el eje “Z” se tiene:

$$R_Z = \sum_{i=1}^n W_{\text{eslabon } i} = 1,23 \text{ N} + 1,98 \text{ N} + 0,89 \text{ N} + 0,90 \text{ N} + 0,67 \text{ N} + 0,49 \text{ N}$$

$$R_Z = 6.16 \text{ N}$$

Aplicando sumatoria de momentos en ambos ejes de tiene:

$$\sum M_X = 0 \quad \text{Ec. 3.6}$$

$$M_X - 0,0701 - 0,2327 - 0,1209 - 0,1077 - 0,0787 - 0,0579 = 0$$

$$M_X = 0,6680 \text{ N} \cdot \text{m}$$

$$\sum M_Y = 0 \quad \text{Ec. 3.7}$$

$$M_Y - 0,2297 - 0,2306 - 0,2331 - 0,1901 - 0,1464 = 0$$

$$M_Y = 1 \text{ N} \cdot \text{m}$$

Entonces, aplicando el Teorema de Pitágoras, el momento resultante es igual a:

$$M_R = \sqrt{(0,6680)^2 + (1)^2} = 1.2026 \text{ N} \cdot \text{m} \quad \text{Ec. 3.8}$$

Con la cinética de cuerpos rígidos - leyes de Newton - se determina el peso necesario de la base del robot para evitar el fenómeno de volcadura; para ello se iguala la suma de momentos estáticos con la suma de momentos cinéticos.

El momento cinético es el producto del momento de inercia respecto a su centro de gravedad y su aceleración angular, cuando éste se ubica en el origen de coordenadas en su centro de gravedad:

$$\Sigma M_{CG} = I_{CG} * \alpha \quad 3.9$$

En este caso la aceleración angular es igual a cero debido a que no se evaluará la velocidad de volcadura, sino el instante antes para que esto suceda, tal como se aprecia en la 3.5.

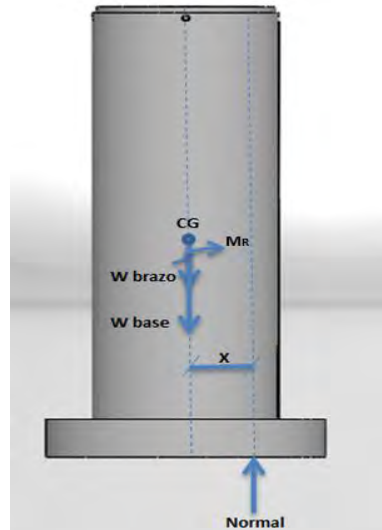


Figura 3.5. Diagrama de cuerpo libre de la base

Fuente; elaboración propia

$$\begin{aligned} N * x - M_R &= 0 \\ N * x &= M_R \\ (W_{Brazo} + W_{Base}) * x &= M_R \\ x &= \frac{M_R}{W_{Brazo} + W_{Base}} \end{aligned}$$

Donde:

- Peso del brazo = $W_{Brazo} = 6.16 \text{ N}$
- Normal = $N = W_{Brazo} + W_{Base}$
- Momento resultante = $M_R = 1.2026 \text{ N} * \text{m}$
- Diámetro del disco base = 0.171 m

$$x = \frac{M_R}{W_{Brazo} + W_{Base}} < \frac{0.171}{2} = 0.0855 \text{ m}$$

$$M_R = 0.0855 * (W_{Brazo} + W_{Base})$$

$$W_{Base} = \frac{M_R - 0.0855 * W_{Brazo}}{0.0855}$$

$$W_{Base} = \frac{1.2026 - 0.0855 * 6.16}{0.0855}$$

$$W_{Base} = 7.9 \text{ kg}$$

3.2.3 Selección de los componentes principales

3.2.3.1 Encoder Incremental

Como se describió en el primer capítulo, los tipos de encoder más utilizados en la industria son: incrementales y absolutos. Para la aplicación de escaneo, ambos resultarían de utilidad ya que permitirían obtener los desplazamientos de tipo angular de los eslabones, con la diferencia de que el encoder absoluto presenta una referencia fija y cada posición angular le corresponde un único mensaje digital, de tal manera que si la energía se pierde, su posición angular será corregida cada vez que la energía sea reestablecida, mientras que para los encoders incrementales su referencia no es fija, es decir, su referencia se establece cuando se le suministre energía independientemente de la posición en la que se encuentre, a partir de ahí, empieza el incremento o disminución de ángulos.

El encoder incremental es el componente fundamental de cada articulación del brazo robot porque permite determinar los desplazamientos de tipo angular de cada uno de eslabones del brazo robot respecto a una posición inicial o de referencia, cuando se realiza el proceso de escaneo. El tipo elegido fue un encoder incremental con resolución de 1200 pulsos/revolución por su disponibilidad física en el Laboratorio de Ingeniería Mecánico Eléctrica de la Universidad de Piura, y por tener una resolución aceptable para el escaneo de curvas, donde cada pulso representa 0.3 grados de recorrido angular.

El encoder incremental seleccionado fue el GHS38 que corresponde a 1200 pulsos/revolución, el mismo que se muestra en la figura 3.7. Sus datos técnicos se muestran en el anexo C.



Figura 3.6 Encoder incremental - GHS38 – 1200 pulsos/revolución.

Fuente: CALT (2015). Shangay Qiyi Electrical and Mecanical Equipment Manual.

3.2.3.2 Placa electrónica

Placa electrónica está basada en la plataforma de hardware Arduino Nano, cuya construcción fue realizada por un técnico electrónico. Su función es la lectura de datos brindados por los Encoder (figura 3.7).

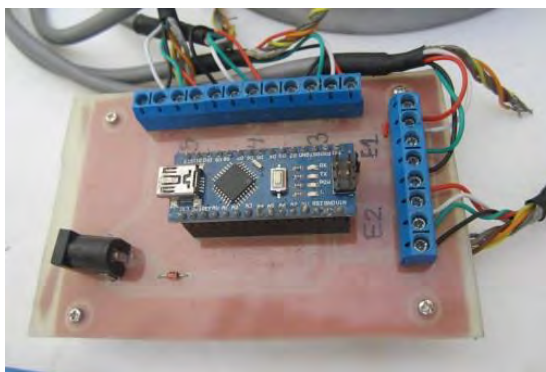


Figura 3.7. Placa electrónica basada en Arduino Nano.

Fuente: Laboratorio de Electrónica. UDEP.

3.2.3.3. Piezas de poliéster ertalyte para los eslabones

La selección del material para fabricar los eslabones del brazo robot cumple los siguientes requisitos:

- Material muy liviano.
- Resistencia y rigidez elevada.
- Dureza de la superficie del material.
- Facilidad para ser torneado y pulido

Para satisfacer las necesidades anteriores, se decidió consultar los tipos de materiales disponibles un material con tales características en el laboratorio de Ingeniería Mecánica Eléctrica de la Universidad de Piura, obteniendo como mejor respuesta el material Ertalyte, que además de sus propiedades, se encontraba disponible para su utilización.

El ertalyte es un termoplástico, semicristalino de poliéster no reforzado hecho de polietileno tereftalato (PET-P). Está fabricado con resinas. Se caracteriza por tener la mejor estabilidad dimensional junto con una excelente resistencia al desgaste, un bajo coeficiente de fricción, alta resistencia y resistencia a soluciones moderadamente ácidas. Sus propiedades lo hacen especialmente adecuado para la fabricación de piezas mecánicas de precisión que son capaces de sostener grandes cargas y soportar condiciones de desgaste. Se pueden construir elementos de deslizamiento y de cojinetes, piezas de bombas, piezas de carcasas, ruedas dentadas, piezas aislantes en la electrotecnia, entre otras.

Para obtener las partes del robot, diseñadas previamente, se utilizaron retazos de ertalyte de 12.8 mm de espesor. En la figura 3.8 se muestran los eslabones y piezas de unión diseñadas y construidas con ertalyte,

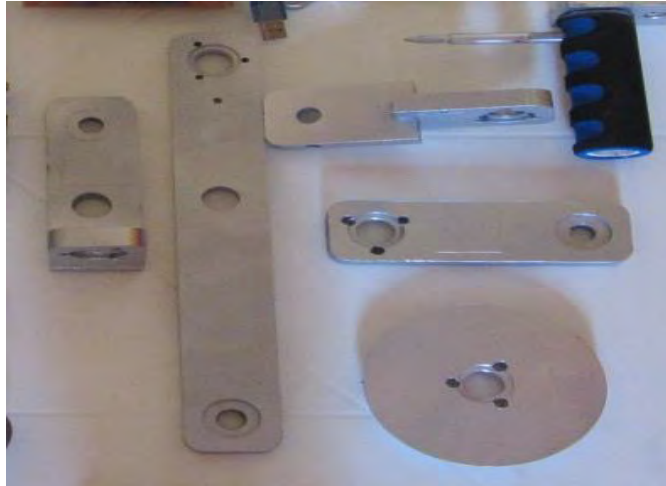


Figura 3.8. Piezas construidas con ertalyte.

Fuente: elaboración propia.

3.2.3.4 Tuerca M12 media rosca

Esta tuerca es utilizada para extender el eje del encoder, funcionando concéntrico a él, ganando espacio para poder compactar los componentes de cada articulación del brazo robot (figura 3.9).

Para su selección se consideró lo siguiente:

- Longitud mínima del perno de 40 mm y no mayor a 60 mm ya que cada articulación comprende la unión de dos eslabones, cada uno de ellos de 12.8 mm de espesor haciendo un primer grosor de articulación de 25.6 mm, sin considerar el espesor de la tuerca ligada al mencionado perno y que compacta cada articulación del brazo robot. Generalmente dichas tuercas presentan un espesor de 10 mm. El perno no debe exceder los 60 mm ya que sobraría parte del perno siendo visible y no estético para cada articulación.

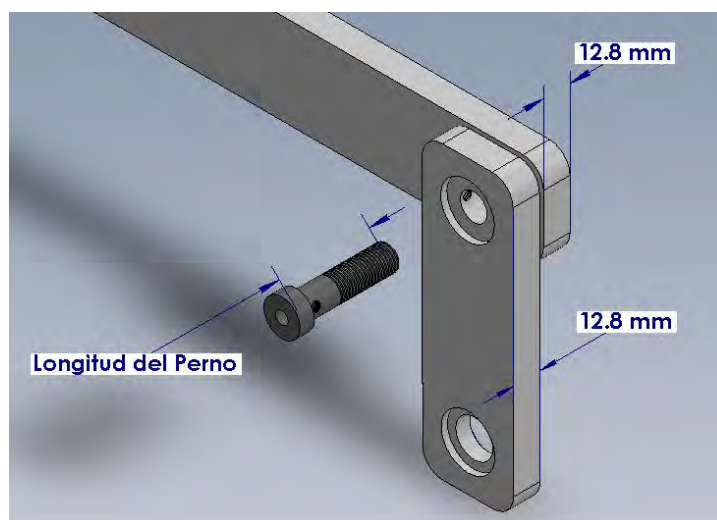


Figura 3.9. Montaje de perno de sujeción.

Fuente: elaboración propia.

- El eje del encoder entra en el perno concéntricamente, para ello se hace un agujero en la cabeza del perno de diámetro igual al diámetro del eje del encoder, con ciertas tolerancias y con una profundidad de 15 mm. Para evitar el deslizamiento entre estas dos partes se utiliza un prisionero, así se evita el deslizamiento entre el perno y el eje del encoder (figura 3.10).

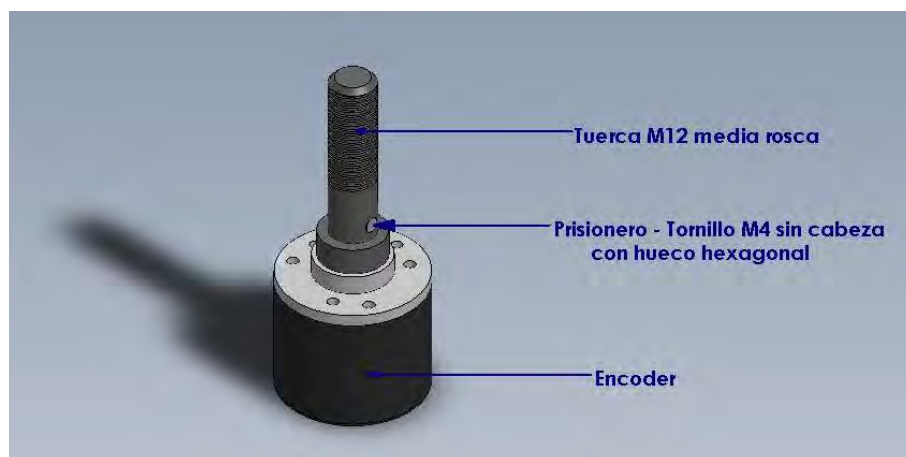
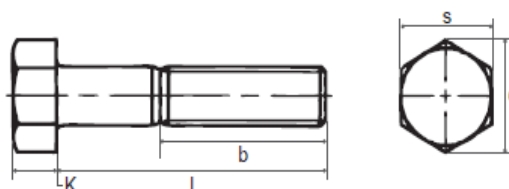


Figura 3.10. Instalación del encoder en el perno.

Fuente: elaboración propia.

- Considerando estas dos características, la mejor selección es un perno M12 media rosca con una longitud de 50 mm, 30 mm de parte roscada y 20 mm sin roscar con diámetro igual a 11 milímetros, según catálogo técnico de pernos y tuercas (figura 3.11).



d	M4	M5	M6	M8	M10	M12	M14	M16	M18	M20	M22	M24
s	7	8	10	13	17	19	22	24	27	30	32	36
k	2.8	3.5	4	5.3	6.4	7.5	8.8	10	11.5	12.5	14	15
Paso	0.7	0.8	1	1.25	1.5	1.75	2	2	2.5	2.5	2.5	3
b(l<=125)	14	16	18	22	26	30	34	38	42	46	50	54
e	7.66	8.79	11.05	14.30	18.90	21.1	24.49	26.75	30.14	33.53	35.72	39.98

Figura 3.11. Selección de un perno comercial

Fuente: Catálogo de tornillería y remaches.

3.2.3.5 Rodamiento 51101

Para soportar las cargas axiales y suavizar el movimiento relativo presente en cada una de las articulaciones del brazo robot.

Para su selección se consideró lo siguiente:

- Según la configuración y el diseño del brazo robot, el rodamiento debe ser de tipo axial para soportar las cargas axiales al momento de asegura la tuerca con el perno, regulando así el movimiento relativo en cada una de las articulaciones (figura 3.12).
- Los diámetros internos de ambas pistas el rodamiento, deben ser mayores o iguales al diámetro externo del perno M12 igual a 12 mm.
- La velocidad de funcionamiento para el rodamiento se considera despreciable debido a que el proceso de escaneo se considera lento.
- La mayor carga de trabajo vendrá soportada por el rodamiento de la “unión base”, cuya carga fue calculada en el análisis estático.
- La mejor selección para soportar una carga de 11.9 kilos es un rodamiento axial 51101, cuyas características se muestran en la figura 3.13.

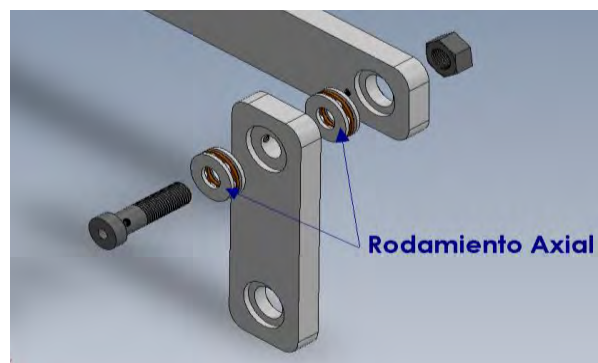


Figura 3.12. Montaje de rodamientos.

Fuente: elaboración propia.

Principal dimensions			Basic load ratings		Fatigue	Minimum	Speed ratings		Mass	Designation
			dynamic	Static	load limit	load factor	Reference speed	Limiting speed		
d	D	H	C	Co	Pu	A				
mm			kN		kN	-	r/min		kg	-
12	26	9	10.4	16.6	0.62	0.0014	9000	13000	0.022	51101

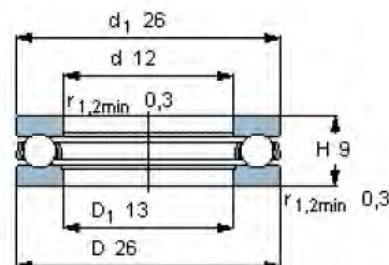


Figura 3.13. Selección de rodamiento.

Fuente: Catálogo de rodamientos SKF

3.2.4 Componentes secundarios

Los componentes secundarios utilizados en el montaje del brazo robot son:

- Disco base: disco de hierro fundido donde se montará el cilindro para juntos formar parte de la base del robot.
- Cilindro base: cilindro que encaja a presión dentro del disco base.
- Tuerca hexagonal M12: solidaria al tornillo M12, su función es recibir el torque de ajuste en las articulaciones para regular el deslizamiento entre las piezas del robot.
- Pulsador: pieza de acero obtenida gracias a un proceso de torneado y que será utilizada para estar en contacto con las superficies a escanear.
- Prisionero tornillo M4 sin cabeza con hueco hexagonal de longitud adaptada a 3.5mm: su función es la de evitar el deslizamiento del eje del encoder dentro del tornillo M12.
- Bocina: Su función es soportar los rodamientos cuando se realice el ajuste en cada articulación, a fin de que sea éste quien soporte estas cargas y no las piezas del brazo que son de una menor dureza. Su diámetro interno es igual al diámetro externo del tornillo M12 y su diámetro externo es igual al agujero dentro de la pieza de manera que se ensamble en ella con una pequeña holgura.
- Tornillo M3 de longitud adaptada a 4 mm: su función es fijar el encoder a una plaquita que proporcionará la unión a su respectiva pieza del brazo robot.
- Tornillo M4 de longitud adaptada a 6mm: su función es la de fijar la plaquita de unión a la pieza del brazo robot.
- Tornillos M4 de longitud adaptada a 16 mm, cuatro para fijar la tapa base del robot al cilindro base y cinco para evitar el deslizamiento de los brazos siguientes a cada articulación, logrando que tanto el eje del encoder como el tornillo M12-media rosca y el brazo siguiente a cada articulación, giren el mismo ángulo.
- Plaquita de unión encargada de proporcionar la fijación del encoder con su respectiva pieza.

El detalle de los componentes secundarios se muestra en el anexo D.

3.2.5 Trabajos de mecanizado y ensamble del brazo robot

El disco base fue torneado para que el cilindro base entre a presión en él y así ir formando la base del robot. Posteriormente se torneó una pieza de nylon para formar la tapa base, ésta deberá entrar con holgura dentro del cilindro base y se fijará con cuatro tornillos M4x16. Para conseguirlo se realizaron 4 agujeros en el cilindro base con un taladro con broca 3 mm y luego se utilizó un macho que brinde el roscado necesario para que se puedan montar cuatro tornillos M4x16, tal y como se muestra en la figura 3.14.

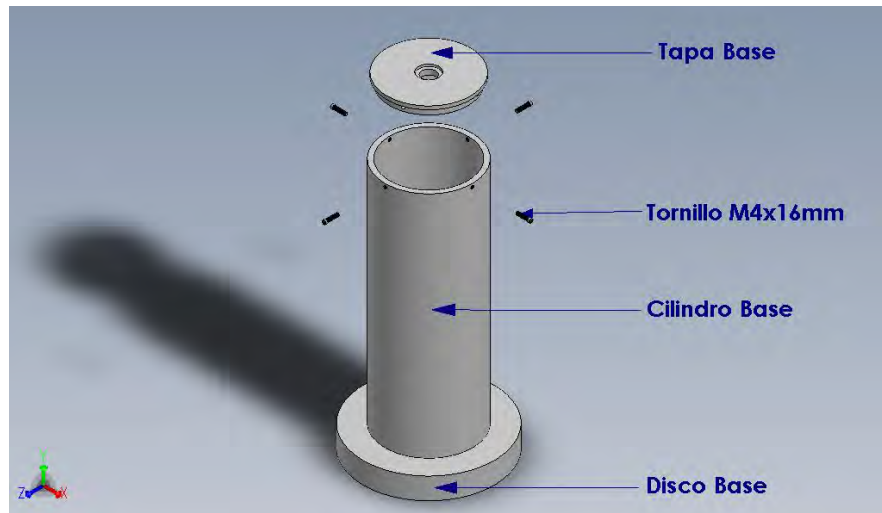


Figura 3.14. Ensamble del disco y tapa base dentro del cilindro base.
Fuente: elaboración propia.

La tapa base también fue sometida a un proceso de rebaje y perforación pasante. El agujero pasante encaja en la tapa base y soporta la presión de los dos rodamientos que se unen a la articulación. Los rebajes en la tapa base fueron realizados concéntricos a su agujero y en ambos lados de la pieza, siendo estos los que sirvan de fijación para los rodajes 51101, tal como se muestra en la figura 3.15.

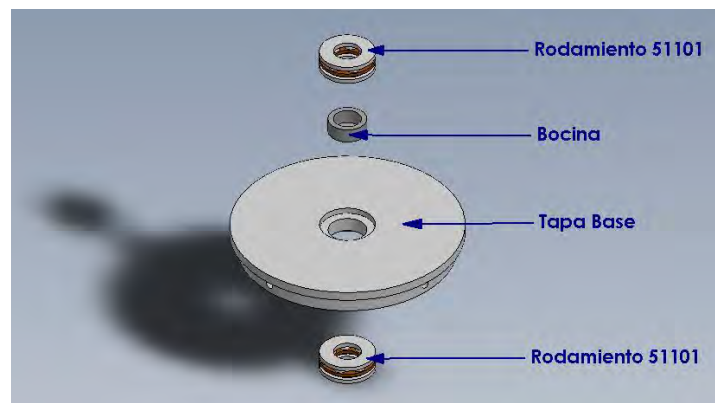


Figura 3.15. Ensamble de rodajes y bocina en tapa base.
Fuente: Elaboración propia.

Se realizó un agujero de 6mm de diámetro y 15 mm de profundidad en los pernos M12 para que el eje del encoder entre dentro de él, para ello se utilizaron brocas de 5mm y 6mm. Luego se le realizó un nuevo agujero roscado en la parte no roscada del perno, primero con broca de 3mm y luego un macho de 4mm que genere el roscado necesario para que un prisionero o tornillo M4x3.5mm se ensamble y evite el deslizamiento entre el eje del encoder y el perno M12 tal y como se aprecia en la figura 3.16.

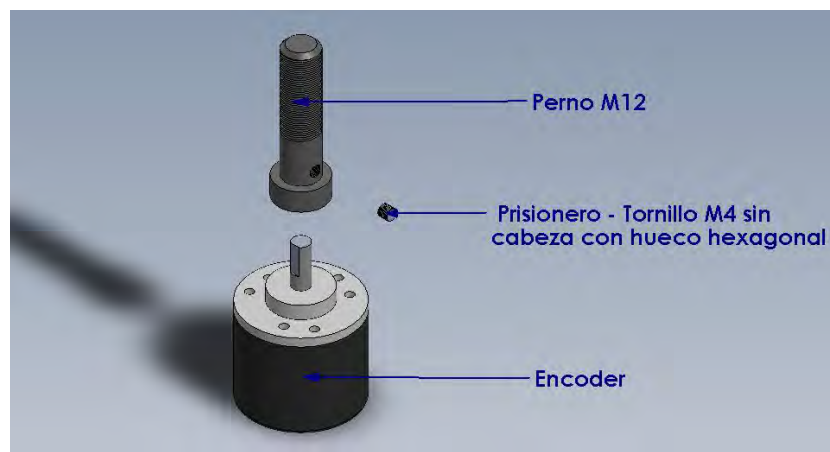


Figura 3.16. Ensamblaje del eje del encoder en el tornillo M12
Fuente: elaboración propia.

Una vez montados los 2 rodajes y la bocina en una determinada articulación, es necesario realizar un agujero pasante para que pase con holgura el tornillo M12 y el mismo rebaje en la pieza siguiente que formará el brazo robot.

La figura 3.17 muestra el brazo 1 y para que no exista deslizamiento, entre el tornillo M12 y el brazo 1, se realiza un agujero en el brazo 1 transversal con broca 3 y luego un roscado con macho 4 para que se pueda montar el tornillo M4x16.

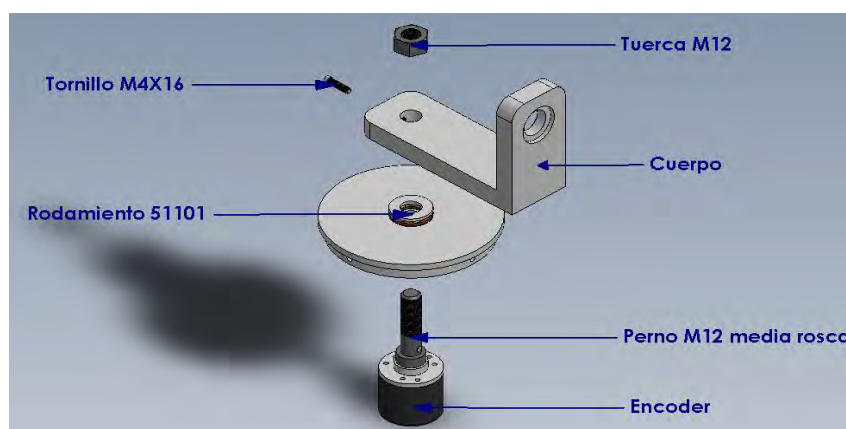


Figura 3.17. Explosionado de piezas en una articulación del brazo robot
Fuente: elaboración Propia.

Todos los procedimientos de trabajo y ensamble se desarrollan en cada una de las articulaciones del brazo robot (figura 3.18), así las cargas y los momentos generados debido al peso de las piezas y componentes del brazo robot, no son aplicadas en el eje del encoder porque se dañaría, sino en el tornillo M12. Este proceso se repite para las articulaciones restantes.

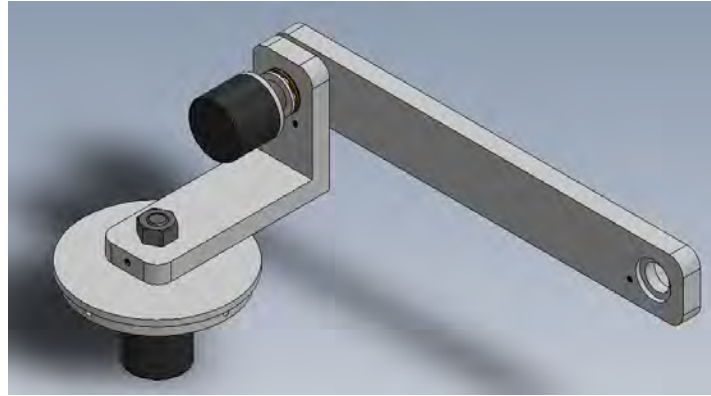


Figura 3.18. Ensamble de las dos primeras articulaciones del brazo robot.
Fuente: elaboración propia.

La figura 3.19 muestra la vista de sección de las articulaciones 1 y 2 junto con sus respectivos encoders, aquí claramente se aprecia como el rebaje realizado en la tapa base y el brazo 1 sirve de apoyo para las pistas del rodamiento, además, se aprecia que el tornillo M4x16 evita el deslizamiento entre el brazo 1 y el tornillo M12x40, el mismo principio es aplicado en las siguientes articulaciones.

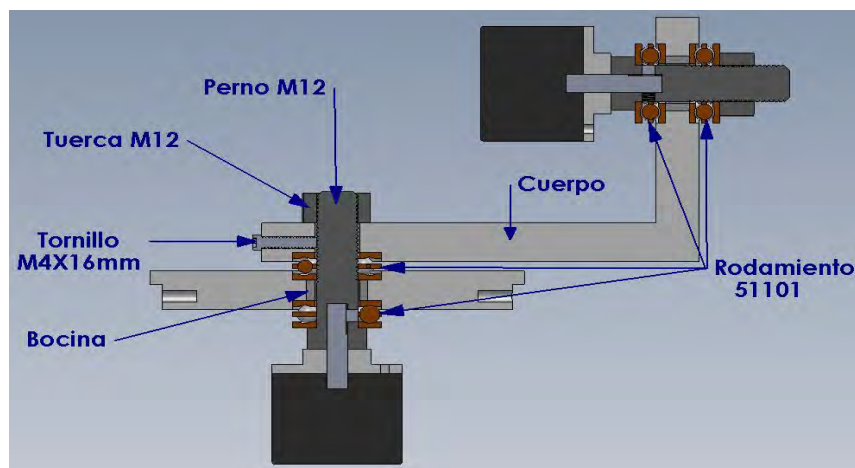


Figura 3.19. Vista de sección de las dos primeras articulaciones.
Fuente: elaboración propia.

La figura 3.20 muestra la última condición para el diseño mecánico de cada una de las articulaciones, la de fijar los encoders a las piezas del brazo robot, para ello es necesario resaltar que el torque para mover el eje es mínimo, basta asegurar el encoder a cada una de las piezas donde funcionará mediante una pequeña pieza llamada “plaquita metálica” para evitar deslizamientos de éste; de esta manera, sólo girará el eje del encoder tomando como dato el ángulo de giro respecto a una posición inicial.

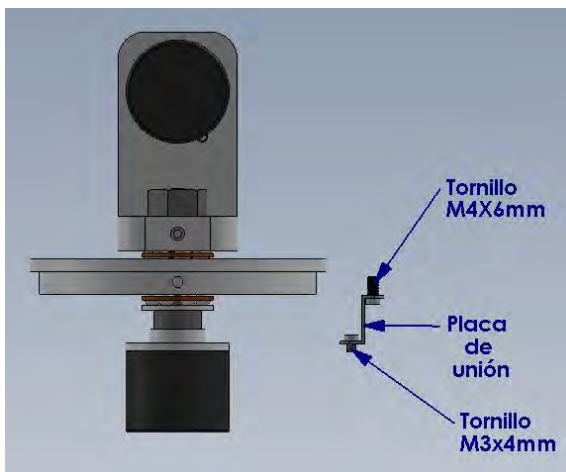


Figura 3.20. Desmontaje de la placa de unión encoder – pieza.
Fuente: elaboración propia.

Finalmente, la figura 3.21 muestra el ensamble final del brazo robot en función al diseño mecánico que se ha considerado para su construcción, el brazo presenta una serie de ventajas como son: estabilidad y buen manejo para una toma correcta de datos.



Figura 3.21. Ensamble final del brazo robot.
Fuente: elaboración propia.

3.2.6. Listado y costo de materiales

La cotización del brazo robot se ha dividido de una cotización de los componentes principales para su desarrollo y la otra que considera las herramientas secundarias a utilizar para el ensamblaje. El detalle de la cotización se presenta en el anexo E.

3.2.7. Especificaciones de trabajo

La altura del cilindro base fue establecida arbitrariamente, ésta puede variar de forma, altura, se puede fijar en forma horizontal o vertical, o se puede establecer una base móvil.

Para efectos del presente estudio se presentan dos posiciones, las más comunes, con sus respectivos rangos de movilidad para cada articulación. Estas especificaciones de trabajo han sido establecidas de acuerdo a las pruebas y los ensayos desarrollados (figuras 3.22, 3.23, 3.24 y 3.25)

1. Posición de trabajo 1.

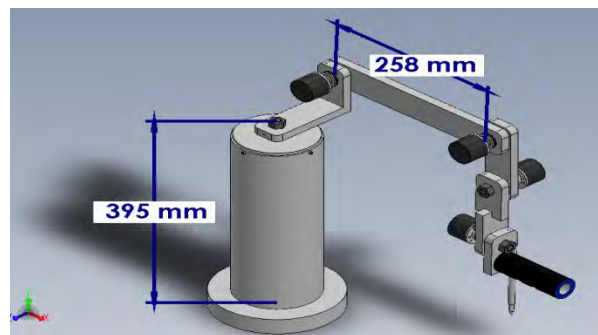


Figura 3.22. Posición de trabajo 1.

Fuente: elaboración propia.

Rangos de trabajo recomendados para la posición de trabajo 1

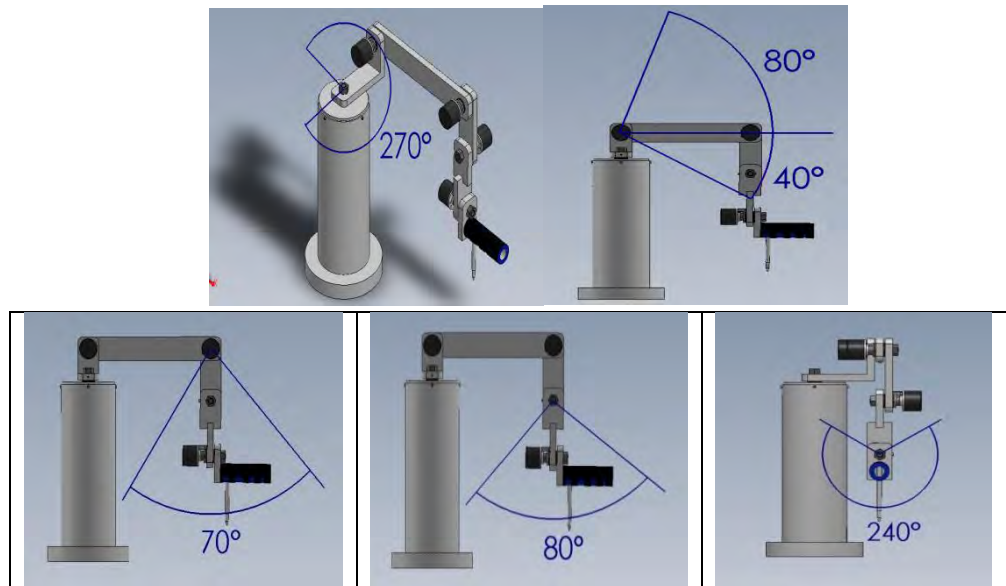


Figura 3.23. Variantes de la posición de trabajo 1.

Fuente: elaboración propia.

2. Posición de trabajo 2.

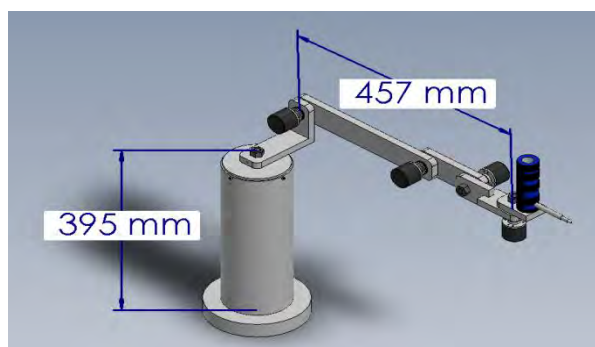


Figura 3.24. Posición de trabajo 2.

Fuente: elaboración propia.

Rangos de trabajo recomendados para la posición de trabajo 2

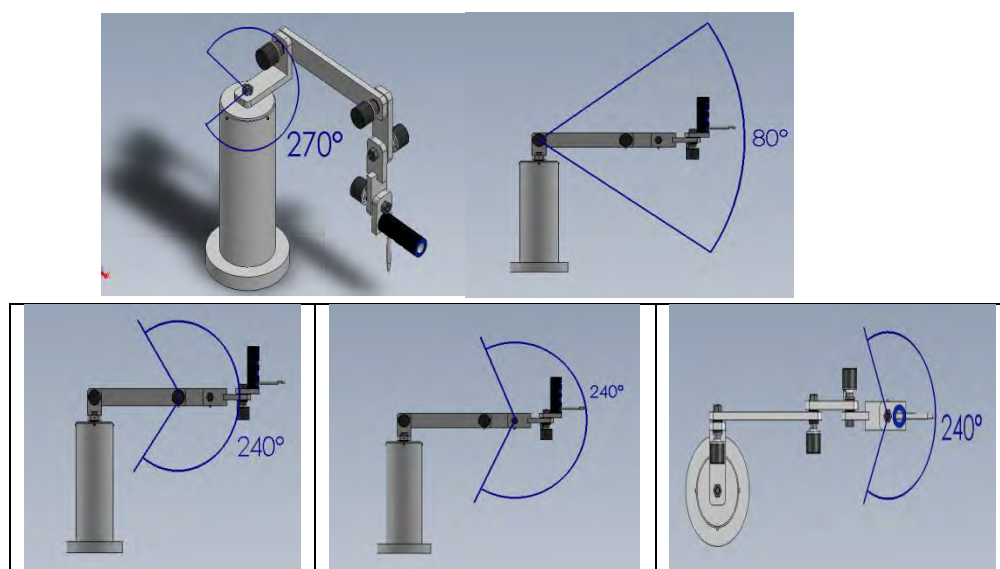


Figura 3.25. Variantes de la posición de trabajo 2.

Fuente: elaboración propia.

Capítulo 4

Programación para el control del brazo robot

4.1 Arduino

Arduino es una plataforma de hardware libre utilizada para desarrollar diversas aplicaciones electrónicas por su facilidad de programación. Se compone de una placa relativamente barata con un micro controlador y un entorno de desarrollo integrado (IDE) encargada de recolectar los datos para su posterior procesamiento.

Es multiplataforma porque se ejecuta en sistemas operativos Windows, Macintosh OSX y GNU/Linux. Posee un entorno de programación simple y claro, de código abierto y software extensible mediante librerías C++, se puede añadir código AVR-C directamente en los programas Arduino.

El Arduino está basado en microcontroladores ATMEGA8 y ATMEGA168 de Atmel. Los planos para los módulos están publicados bajo licencia Creative Commons, por lo que cualquier diseñador experimentados en circuitos puede hacer su propia versión del módulo, extendiéndolo y mejorándolo.

4.1.1 Arduino Nano - Hardware

Para la captura de información de los encoder del brazo robot se empleó Arduino nano ATmega328, una nueva versión de Arduino muy completa y pequeña que se usa conectándola a un protoboard (Tabla 4.1).

Tabla 4.1. Características del protoboard.

Microcontrolador	Atmel ATmega168 o ATmega328
Tensión de operación (Nivel lógico)	5V
Tensión de entrada (Recomendado)	7-12 V
Tensión de entrada (Limites)	6 – 20 V
Pines E/S Digitales	14 (de los cuales 6 proveen de salida PWM)
Entradas analógicas	8
Corriente máxima por cada PIN de E/S	40 mA
Memoria Flash	16 KB (ATmega168) o 32 KB (ATmega328) de los cuales 2KB son usados por el bootloader.
SRAM	1KB (ATmega168) o 2 KB (ATmega328).
EEPROM	512 bytes (ATmega168) o 1 KB (ATmega328).
Frecuencia de reloj	16 MHz.
Dimensiones	18,5 mm x 43,2 mm.

Fuente: Arduino. Recuperado de

<https://www.arduino.cc/en/Main/ArduinoBoardNano>

4.1.2 Software

El entorno de desarrollo de Arduino se denomina IDE (Integrated Development Environment) donde se programa cada placa, basado en “Processing” y otros programas de código abierto.

El código completo realizado en Arduino, para la toma de datos, puede verse en el anexo F; sin embargo, es necesario describir algunos tipos de datos manejados por Arduino, su estructura y funciones importantes.

- Byte: almacena un valor numérico de 8 bits. Tiene un rango de 0 – 255.
- Int: almacena un valor entero de 2 bytes (16 bits), esto supone un rango desde $-2^{15} = -32768$ hasta un valor máximo de $2^{15} - 1 = 32767$
- Long: valor entero almacenado en 4 bytes (32 bits), esto supone un rango entre -2147483648 y 2147483647, utilizado para almacenar variables mayores a las comprendidas en el rango int.
- Float: tipo de datos para números decimales o punto flotante, el valor de las variables tipo float pueden estar en el rango de -3.4028235E+38 a 3.4028235E+38 y se almacenan como 4 bytes.
- Void Setup: parte que inicializa la configuración de un pin o establece la velocidad de comunicación, sólo se ejecuta una sola vez.
- Void Loop: aquí se ejecutan las acciones del programa, una y otra vez línea a línea.
- Pin Mode: configura el pin especificado para comportarse en modo de entrada (input) o modo salida (output).
- Digital Read: lee el valor desde un pin digital específico y devuelve un valor alto (high) o bajo (low).
- Serial Print: imprime los datos al puerto serie como texto legible ASCII. Los datos float son impresos por defecto con dos decimales.

4.1.3 Diagrama de flujo de la programación en Arduino

El diagrama de flujo utilizado para diseñar el código de recolección de datos en Arduino se aprecia en la figura 4.1. Un dato muy importante para su programación fue la resolución del encoder, igual a 1200 pulsos/revolución, porque cada pulso del encoder representa un desplazamiento angular de 0.3 grados.

El código inicia con la declaración de variables, asignación de pines, creación de variables contadoras de pulso y de grados para cada encoder, inicialmente iguales a cero, así como la configuración del tipo de variable a utilizar.

En el “setup”, que representa la preparación del programa, se configuran las entradas mediante el comando pinMode; luego, se inicia el bucle loop o ejecución del programa en sí, el cual incluye el código que será ejecutado continuamente.

Cada vez que se ejecute este bucle se deberán leer los valores digitales de cada pin conectado a cada encoder y almacenar dicha información en variables diferentes para cada encoder.

Los condicionales hacen referencia a la señal de los encoder incrementales y representan el desplazamiento angular de ellos, preguntan si el primer encoder ha cambiado de valor digital respecto a su valor inicial (low), si es verdadero, aparece un nuevo condicional, que hace referencia a la siguiente señal que representa el sentido de giro, horario o antihorario, definiendo el aumento o disminución de grados.

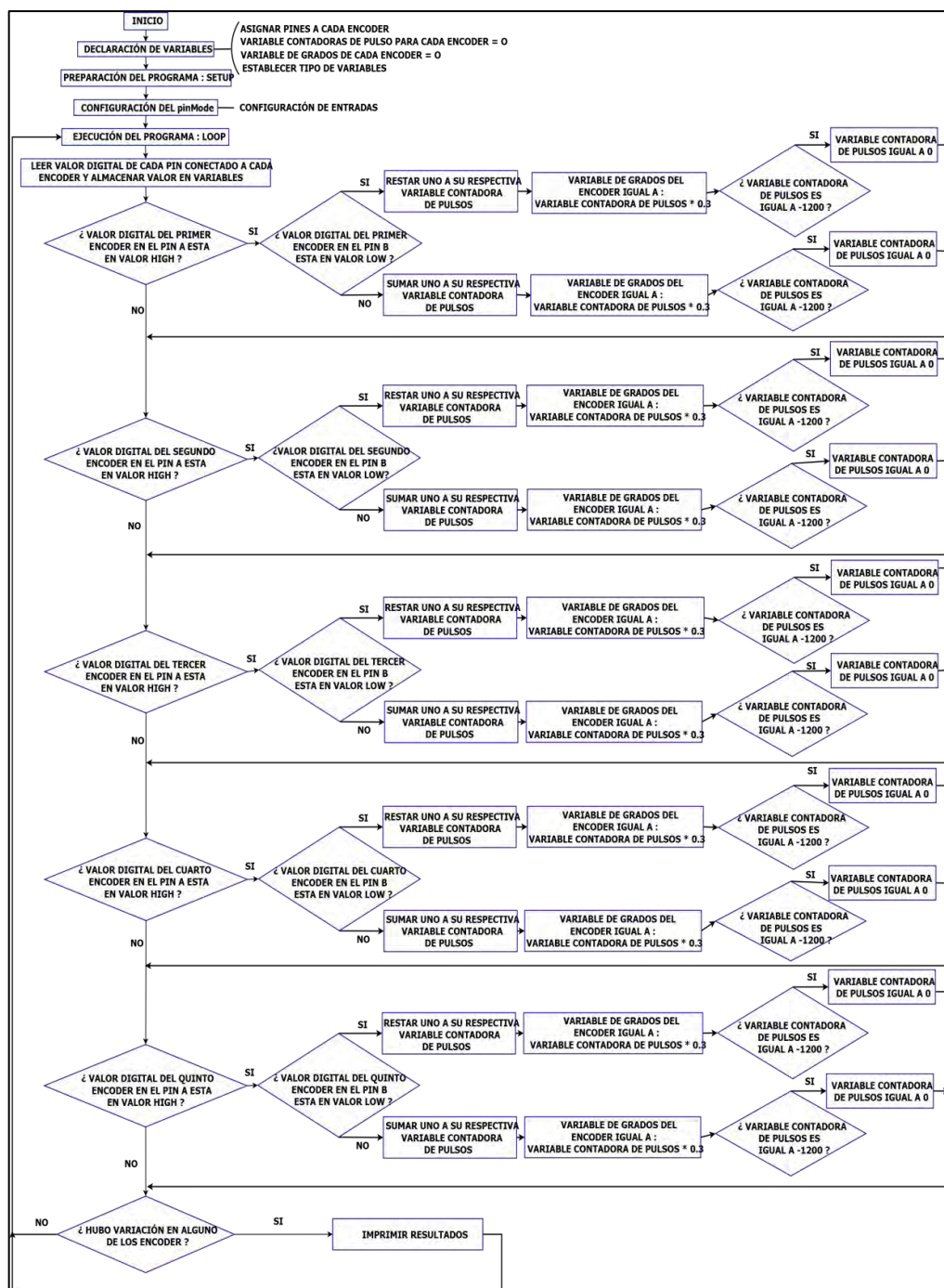


Figura 4.1. Diagrama de flujo para código Arduino.

Fuente: elaboración propia.

Si se toma como ejemplo la primera vez que se ingrese a este condicional con un sentido de giro positivo del eje del encoder, la variable contadora de pulso respectiva que inicia con valor 0, deberá aumentar una unidad debido a que se ha registrado un pulso y deberá actualizar dicho valor, posteriormente la respectiva variable de grados para el primer encoder será igual al valor de la variable contadora de pulsos multiplicada por 0.3, porque un pulso representa 0.3 grados.

Por último, es necesario condicionar el máximo valor de pulsos, en caso llegue a los 1200 pulsos que representan los 360 grados, la variable contadora de pulsos deberán volver a cero. Lo mismo pasa en caso el sentido de giro sea negativo. Todo lo anterior deberá repetirse para cada uno de los 5 encoders.

Cuando se haya terminado de preguntar a cada uno de los encoder por algún cambio de estado, aparecerá un último condicional que determinará si por lo menos hubo un cambio en cualquier encoder, caso verdadero, se deberá imprimir y mostrar el resultado, sino, el programa deberá ejecutarse nuevamente. El resultado obtenido será mostrado como una trama de datos separados por delimitadores “a,b,c,d,e,f” para que, de esta manera, Matlab pueda extraer los ángulos estableciendo como límites sus delimitadores respectivos. Esta sección será explicada a mayor detalle en la programación Matlab.

El desarrollo de este algoritmo de trabajo captura la información; sin embargo, ante cambios de estado simultáneos, el código resulta un poco ineficiente porque al ejecutar las líneas de código para un cambio determinado imprimiendo resultados, simultáneamente ocurre el cambio de posición, originando pérdida de información, que se puede minimizar con un tiempo de espera. Estos inconvenientes serán visualizados con mayor claridad en el capítulo 5 de realización de ensayos.

4.2 Matlab

Matlab es un entorno interactivo para desarrollar algoritmos; analizar y visualizar datos; y realizar cálculos numéricos con un lenguaje de programación propio. Esta herramienta será utilizada para procesar la información capturada por Arduino.

4.2.1 Código base de programación

Inicialmente se diseñó en Matlab un código que permite poner en práctica la matriz de transformación homogénea para calcular las coordenadas en el espacio, a partir de los ángulos de cada articulación y las dimensiones del brazo robot diseñado. Este código es mostrado en la figura 4.2.

De acuerdo a los ejemplos vistos en el modelo cinemático directo del brazo robot, capítulo 2, en el caso inicial, donde todos los ángulos de los encoders son iguales a cero, el código permite obtener la coordenada en el espacio del extremo del pulsador para esos valores.

```

%PROGRAMA DE RESOLUCIÓN MATRIZ DENAVIT - HARTENBERG
clc
clear all
%Configuración de ángulos predeterminados por el usuario
a(1,1)=0;
a(2,1)=0;
a(3,1)=0;
a(4,1)=0;
a(5,1)=0;
a(6,1)=0;

%Parámetros fijos
a(1,2)=46;   a(1,3)=0;   a(1,4)=0;
a(2,2)=0;    a(2,3)=0;   a(2,4)=-pi/2;
a(3,2)=115;  a(3,3)=258;  a(3,4)=0;
a(4,2)=0;    a(4,3)=97;   a(4,4)=0;
a(5,2)=0;    a(5,3)=102;  a(5,4)=pi/2;
a(6,2)=37;   a(6,3)=122;  a(6,4)=0;

MatrizFinal=1;
for i=1:1:6
    r(1,1)=cos(a(i,1));
    r(1,2)=-sin(a(i,1))*cos(a(i,4));
    r(1,3)=sin(a(i,1))*sin(a(i,4));
    r(1,4)=a(i,3)*cos(a(i,1));
    r(2,1)=sin(a(i,1));
    r(2,2)=cos(a(i,1))*cos(a(i,4));
    r(2,3)=-cos(a(i,1))*sin(a(i,4));
    r(2,4)=a(i,3)*sin(a(i,1));
    r(3,1)=0;
    r(3,2)=sin(a(i,4));
    r(3,3)=cos(a(i,4));
    r(3,4)=a(i,2);
    r(4,1)=0;
    r(4,2)=0;
    r(4,3)=0;
    r(4,4)=1;
    H=[r(1,1) r(1,2) r(1,3) r(1,4);r(2,1) r(2,2) r(2,3)
        r(2,4);r(3,1) r(3,2) r(3,3) r(3,4);r(4,1) r(4,2)
        r(4,3) r(4,4)];
    MatrizFinal=MatrizFinal*H;
end
fprintf('\n')
disp('COORDENADAS PUNTO P')
P=[MatrizFinal(1,4) MatrizFinal(2,4) MatrizFinal(3,4)]

```

Figura 4.2. Código inicial desarrollado en Matlab.

Fuente: elaboración propia.

La coordenada resultantes con el programa son las siguientes: P= 579 115 83, las mismas que se calcularon con el algoritmo D-H para esta posición inicial.

4.2.2 Diagrama de flujo de la programación de la GUI.

Establecido el correcto funcionamiento del código inicial para el cálculo de coordenadas en el espacio, se procede a diseñar un diagrama de flujo como se muestra en la figura 4.3, que representará el diseño de la GUI (Graphic User Interface) para obtener una comunicación usuario - innterfaz.

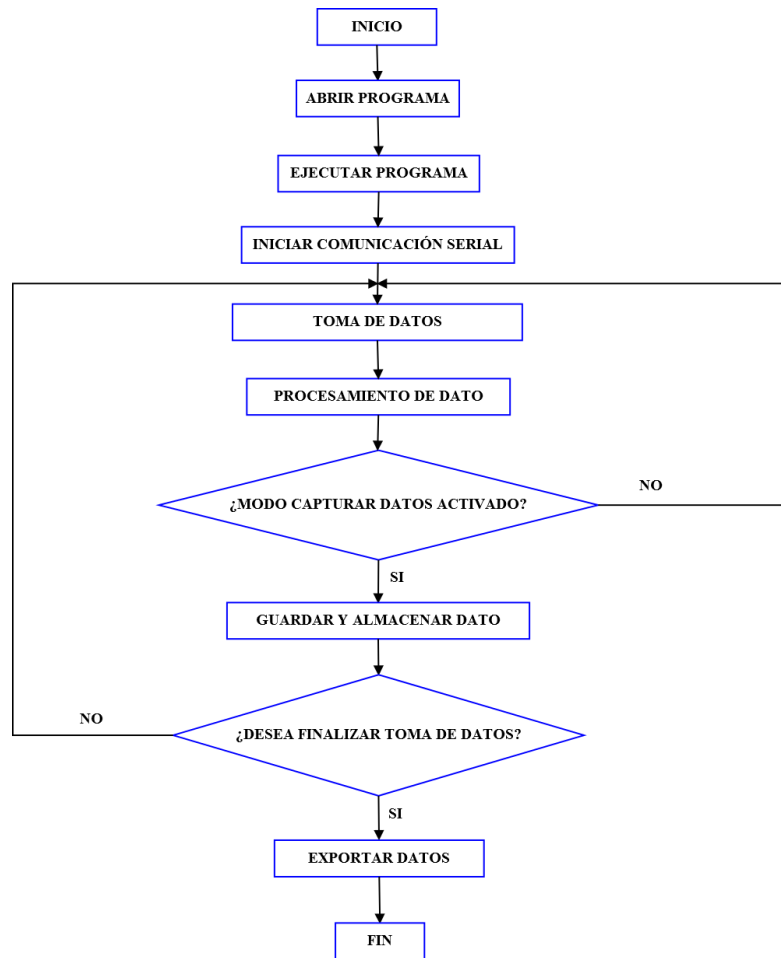


Figura 4.3. Diagrama de flujo código Matlab.
Fuente: elaboración propia.

4.2.3 Descripción de la GUI

Para el diseño de la GUI (Graphic User Interface) en Matlab, es importante mencionar y describir brevemente algunos comandos importantes:

- *Handles*: estructura general de una GUI que contiene objetos.
- *Callback*: subrutina o líneas de código que se ejecutan debido a un evento.
- *Evento*: cualquier interacción con usuario y programa que sea de interés especial, ya sea mediante un click o cualquier otro tipo de interacción que genere una acción dentro de la GUI.
- *Variables*: localidad y globalidad de variables.
- *Objeto*: botones, cajas de selección, texto, entre otros; que son la base para el diseño de una GUI.

La figura 4.4 relaciona de manera general, los comandos mencionados anteriormente. Se aprecia el comando Handles como una estructura general y que dentro de ella puede contener objetos, cada uno de ellos con características propias.

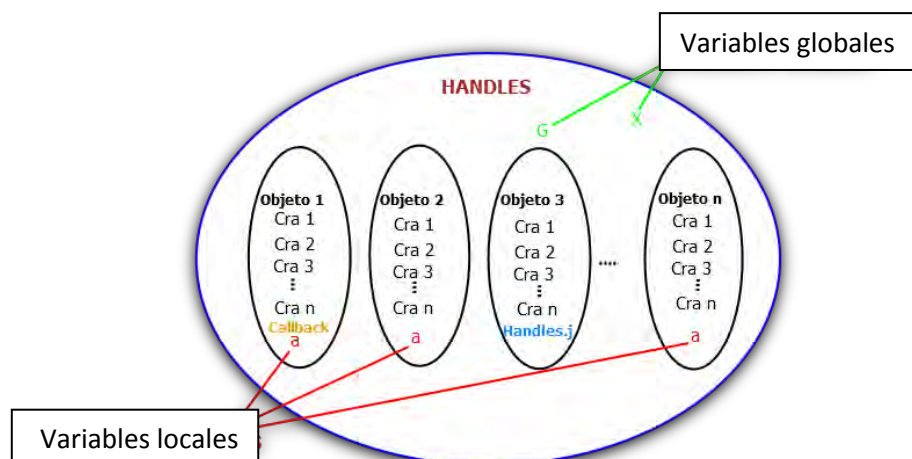


Figura 4.4. Comandos importantes para el diseño de una GUI.

Fuente: Franco R., (2014, 9 de Diciembre). GUI-Interfaz Gráfica de Usuario en Matlab. Recuperado de <https://www.youtube.com/watch?v=jBecrEnkSV0>

El objeto 1 tiene un Callback, o mejor dicho, tiene una sección de código que se ejecutará cuando se ejecute un evento usuario – interfaz. Los objetos pueden o no tener Callback.

Tanto el objeto 1, 2 y “n”, tienen una variable local “a” que puede ser utilizada únicamente dentro del objeto al cual correspondan, además, la variable “a” dentro del objeto 1 es diferente a la variable “a” del objeto 2 y del objeto n, es por eso que se les llama variables locales. Por otro lado, las variables globales G y X, pueden ser utilizadas por cualquiera de los objetos. Su utilización requiere un mayor cuidado, debido a que pueden existir confusiones con otras variables al momento de ejecutar la programación.

El objeto 3 tiene una variable “j” que utilizada como “handles.j” genera una globalización de dicha variable para poder ser utilizada por cualquier otro objeto.

Los diferentes objetos dentro de la GUI se muestran a continuación y pueden encontrarse en la sección mostrada en la figura 4.5. Además, se ha seleccionado un objeto en especial “Push Button” y las características propias que se obtienen al dar doble click sobre el objeto.

- Push Button (Botón de presión)
- Slider Bar (Barra deslizante)
- Radio Button (Botón circular)
- Check Box (Caja de selección)
- Edit Text (Texto editable)
- Static Text (Texto no editable)
- Pop-up Menu (Menú desplegable)
- List Box (Caja de lista)
- Toggle Button (Botón de estado)
- Axes (Gráficos)
- Panel (Paneles de botones)

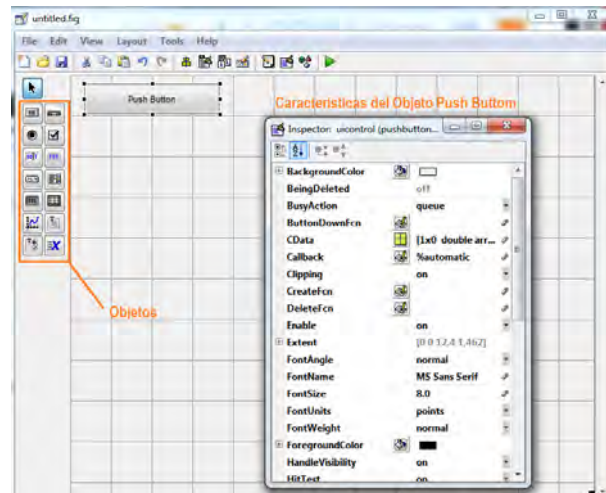


Figura 4.5. Sección de objetos dentro de la GUI.
Fuente: Interfaz gráfica de Matlab..

Para visualizar el código que permite variar las características del objeto frente a un evento usuario-interfaz se deberá dar anticlick en el objeto y seguir las indicaciones mostradas en la figura 4.6.

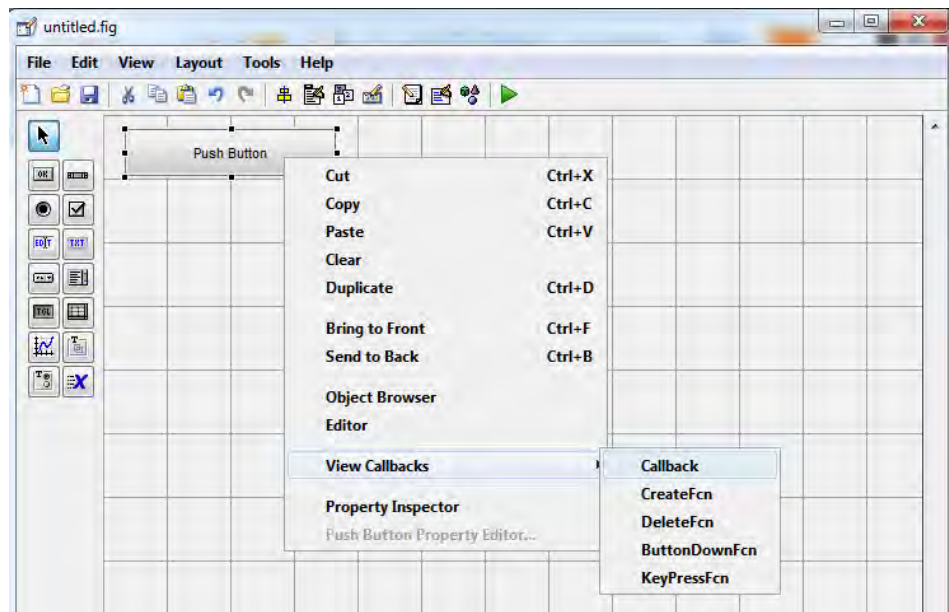


Figura 4.6. Pasos para visualizar el código de programación del objeto.
Fuente: Interfaz gráfica de Matlab..

Aquí se muestra el diseño de la GUI utilizada para el procesamiento de la información recogida por Arduino al tomar los ángulos de giro de las articulaciones del brazo robot, también detalla cada objeto con su respectivo Tag (característica de los objetos que representa su nombre dentro de la programación).

Además, se aprecia que los únicos parámetros a ingresar en la GUI son el puerto de comunicación y la velocidad de transmisión de datos, que para este caso es igual a 11520 baudios.

En caso se desconozca el puerto de comunicación, se puede averiguar siguiendo la secuencia en el computador una vez conectado el cable de transmisión de datos Arduino – PC.

La figura 4.7 muestra el inicio, donde se encuentra el panel de control-hardware y sonido, dispositivos e impresoras, administrador de dispositivos-puertos (COM y LPT), puerto COM.



Figura 4.7. Diseño de la GUI – Objetos utilizados con su respectivo Tag.

Fuente: Interfaz gráfica de Matlab.

El código completo de la GUI se presenta en el anexo G. Las funciones mayormente utilizadas en la programación fueron:

- Los comandos "hObject" o "handles.Nombredelobjeto" son utilizados para referir al objeto en el cual se encuentran.
- "get" es utilizado para tomar una característica del objeto.
- "set" se utiliza para mostrar una característica de un objeto.

Por ejemplo, si se abre el Callback del objeto "pushbutton1_Callback" y se desea variar y mostrar la característica "FontSize" con un valor igual a 15, se utilizará la función "set" según lo explicado anteriormente. Pero, como la característica a variar es propia del objeto mencionado, es necesario referirse a él

mediante “hObject” o “handles.pushbutton1” tal como se muestra en las figuras 4.8 y 4.9 respectivamente.

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
set(hObject, 'FontSize', 15);
```

Figura 4.8. Modificación 1 de la característica de un objeto.

Fuente: Interfaz gráfica de Matlab.

```
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
set(handles.pushbutton1, 'FontSize', 15);
```

Figura 4.9. Modificación 2 de la característica de un objeto.

Fuente: Interfaz gráfica de Matlab..

La figura 4.10 muestra las primeras líneas del código de la interfaz gráfica del brazo robot generadas por defecto por Matlab para inicializar la GUI.

```
% --- CÓDIGO DE INICIALIZACIÓN
function varargout = gui_rx(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @gui_rx_OpeningFcn, ...
    'gui_OutputFcn',  @gui_rx_OutputFcn, ...
    'gui_LayoutFcn',   [] , ...
    'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

Figura 4.10. Líneas de código de inicialización de la GUI.

Fuente: Interfaz gráfica de Matlab..

En la figura 4.11 aparecen las líneas de código generadas por Matlab para establecer las condiciones iniciales del programa.

```

% --- CONDICIONES INICIALES DEL PROGRAMA
function gui_rx_OpeningFcn(hObject, eventdata, handles, varargin)
% Centrar la GUI
movegui(hObject, 'center')
% Deshabilitar botón de envío serial y parada del bucle
set(handles.rx_b, 'Enable', 'off')
set(handles.stop_b, 'Enable', 'off')
handles.acc=[ ];
% Choose default command line output for gui_rx
handles.output = hObject;
% Update handles structure

```

Figura 4.11. Condiciones iniciales generadas por Matlab.

Fuente: Interfaz gráfica de Matlab.

En esta sección se han añadido las condiciones iniciales adicionales de la GUI del brazo robot y se detallan en la figura 4.12. Se puede apreciar el uso del comando set para mostrar una característica del objeto en el cual se encuentra, el uso de “handles.rx_b” o “handles.stop_b” para hacer referencia al objeto cuyo Tag es “rx_b” y “stop_b” respectivamente.

La finalidad de las condiciones iniciales añadidas es la de modificar la característica “enable” de los objetos, la cual indica una habilitación o no del “button” respectivo, es por ello que el “enable” en estado “off” significa una inhabilitación del “button”, en otras palabras, no se realizará ninguna acción o ningún evento cuando se hace un “click” sobre dicho objeto. Mientras que “handles.acc=[]” indica la concatenación de datos dentro de la GUI.

```

set(handles.rx_b, 'Enable', 'off')
set(handles.stop_b, 'Enable', 'off')
handles.acc=[ ];

```

Figura 4.12. Condiciones iniciales de la GUI del brazo robot.

Fuente: elaboración propia.

Después de establecer las condiciones iniciales, se debe iniciar la comunicación serial Arduino – Matlab para poder realizar el procesamiento de los datos. Las líneas de código para habilitar el puerto serial, una vez ejecutado un evento sobre el button “Establecer Comunicación”, son las mostradas en la figura 4.13.

```

% --- ABRIR PUERTO SERIAL.
function com_ser_Callback(hObject, eventdata, handles)
% Desactivar botón actual
set(hObject, 'Enable', 'off')
pause(0.1)
% Tomar y verificar el ID del puerto serial
com=get(handles.com_e, 'String');
if isempty(com) || isnan(str2double(com))
    return
end
% Obtener velocidad (baudrate)
contents = get(handles.los_baud, 'String');
vel=str2double(contents(get(handles.los_baud, 'Value')));
% Configuración del puerto serial
handles.SerPIC = serial(['COM',com]);%Seleccionar puerto COM
set(handles.SerPIC, 'BaudRate', vel);%Velocidad en baudios
set(handles.SerPIC, 'DataBits', 8);%8 bits de datos
set(handles.SerPIC, 'Parity', 'none');%Sin control de paridad
set(handles.SerPIC, 'StopBits', 1);%Un bit de parada
set(handles.SerPIC, 'FlowControl', 'none');%Sin control de flujo
set(handles.SerPIC, 'Terminator', 13);%Terminador
set(handles.SerPIC, 'TimeOut', 2);%Tiempo de espera
% Abrir el puerto serial
fopen(handles.SerPIC);
% Mensaje de apertura del puerto serial
msgbox(['El Puerto COM',com,' está activado'],'Comunicación Serial')
% Habilitar botón de envío
set(handles.rx_b, 'Enable', 'on')
set(handles.stop_b, 'Enable', 'on')
% Actualizar estructura handles
guidata(hObject, handles)

```

Figura 4.13. Código para establecer comunicación serial Arduino – Matlab
Fuente: elaboración propia.

El button “Iniciar Comunicación” tiene como “Tag” el nombre de “com_ser”, por lo tanto, una vez realizado el evento se deberá deshabilitar su “enable” mediante el comando “set(hObject, 'Enable', 'off')”.

Además, el valor colocado en el “Static Text”, cuyo “Tag” es “com_e” y representa el puerto serial donde se encuentra conectado la placa electrónica al computador, se guardará en la variable “com” para ser utilizado posteriormente dentro de la configuración serial.

Luego se almacena dentro de la variable “contents” el valor de los baudios seleccionados, pero al ser esta una cadena de caracteres es necesario convertirla a un número mediante el comando “str2double”. Luego se almacena dentro de la variable “vel”.

Posteriormente las líneas de código para terminar la comunicación serial Arduino – Matlab utilizando los datos ingresados son las mostradas en la figura 4.14.

```
% Configuración del puerto serial
handles.SerPIC = serial(['COM',com]);%Seleccionar puerto COM
set(handles.SerPIC,'BaudRate',vel);%Velocidad en baudios
set(handles.SerPIC,'DataBits',8);%8 bits de datos
set(handles.SerPIC,'Parity','none');%Sin control de paridad
set(handles.SerPIC,'StopBits',1);%Un bit de parada
set(handles.SerPIC,'FlowControl','none');%Sin control de flujo
set(handles.SerPIC,'Terminator',13);%Terminador
set(handles.SerPIC,'TimeOut',2);%Tiempo de espera
% Abrir el puerto serial
fopen(handles.SerPIC);
```

Figura 4.14. Código para establecer comunicación serial Arduino – Matlab.

Fuente: elaboración propia.

Ejecutadas todas las líneas de código anteriores, ante una comunicación serial realizada con éxito, se ha diseñado la aparición de un mensaje que lo indique, utilizando el comando “msgbox”, figura 4.15.

```
% Mensaje de apertura del puerto serial
msgbox(['El Puerto COM',com,' está activado'],'Comunicación Serial')

Además, habilitaremos los button Iniciar Comunicación y Finalizar comunicación cuyo Enable se encontraba en modo Off.
% Habilitar botón de envío
set(handles.rx_b,'Enable','on')
```

Figura 4.15. Mensaje para una comunicación serial exitosa.

Fuente: elaboración propia.

Las líneas de código para iniciar el programa, establecer las condiciones iniciales y la comunicación serial son de gran importancia para comprender el funcionamiento de la GUI.

Posteriormente aparece la programación del código bucle que se ejecutará constantemente y cuya base es el código inicial de la figura 4.2 diseñado en Matlab. Dentro de él se lee continuamente la trama de datos enviada por Arduino, y tal como se mencionó, los ángulos en dicha trama aparecerán separados por delimitadores “a,b,c,d,e,f”, es decir, el ángulo del primer encoder estará entre los delimitadores “a” y “b”, el ángulo del segundo encoder estará entre los delimitadores “b” y “c” y así sucesivamente.

Una vez extraídos los valores de los ángulos, se deberán almacenar dichos valores en variables diferentes para cada encoder, tal como se muestra en las líneas de código de la figura 4.16.

Además, se han añadido las líneas de código que permiten visualizar la trama de datos que imprime arduino en la sección “Datos Recibidos” así como líneas de código que representan tridimensionalmente los puntos calculados xyz en la sección “Gráfica de Puntos Capturados”, su respectivo almacenamiento y la visualización de dichos puntos en la sección “Puntos capturados”.

```

inicio=strfind(handles.acc, 'a');
primero = strfind(handles.acc, 'b');
segundo = strfind(handles.acc, 'c');
tercero = strfind(handles.acc, 'd');
cuarto = strfind(handles.acc, 'e');
fin = strfind(handles.acc, 'f')      ;

t1 = str2double(handles.acc(inicio+1:primero-1));
t2 = str2double(handles.acc(primero+1:segundo-1));
t3 = str2double(handles.acc(segundo+1:tercero-1));
t4 = str2double(handles.acc(tercero+1:cuarto-1));
t5 = str2double(handles.acc(cuarto+1:fin-1));

a(1,1)=degtorad(t1);%encoder 1
a(2,1)=degtorad(t1);%encoder 1
a(3,1)=degtorad(t2);%encoder 2
a(4,1)=degtorad(t3);%encoder 3
a(5,1)=degtorad(t4);%encoder 4
a(6,1)=degtorad(t5);%encoder 5

```

Figura 4.16. Código para extraer los ángulos de la trama de datos de Arduino.
Fuente: elaboración propia.

Capítulo 5

Manual de uso del brazo robot y realización de ensayos

5.1 Guía de usuario

La presente guía de usuario, describirá cada uno de los botones de la interfaz gráfica en Matlab así como los pasos a seguir para poner en funcionamiento el brazo robot y realizar el proceso de escaneo de curvas en 3D.

5.1.1 Descripción de los botones de la GUI

A continuación se describirán cada uno de los botones y secciones dentro de la GUI para una mejor comprensión y correcto uso de la interfaz gráfica de comunicación Usuario-Computador (figura 5.1).

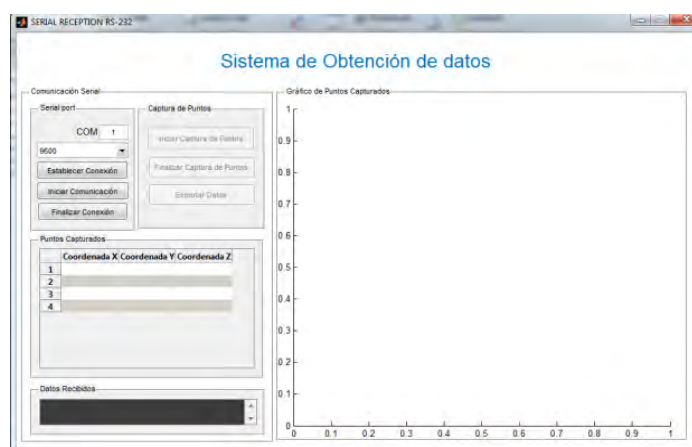


Figura 5.1. Interfaz gráfica para el procesamiento de datos en Matlab.
Fuente: Matlab

- a) **Serial Port:** Objeto de tipo “Static Text”. Sección dentro de la interfaz gráfica vinculada al puerto donde se establecerá la transferencia de datos entre Arduino y Matlab, comprende los siguientes botones:
 - a.1) **COM:** Objeto de tipo “Edit Text”. En él se ingresará el puerto de comunicación serial.
 - a.2) **Pop-up Menú:** Menú desplegable donde se encuentran las diferentes velocidades de transferencia de datos en “baudios”. Para efectos de escaneo, se utilizará **11520**.
 - a.3) **Establecer conexión:** Objeto de tipo “Push Button”. Se da click para ejecutar las líneas de código que permitan establecer la comunicación serial.

- a.4) Iniciar comunicación:** Objeto de tipo “Push Button”. Se da click para que los encoder entren en funcionamiento con un ángulo inicial de cero grados, previamente se deberá colocar el brazo robot en una “posición inicial” que será explicada con mayor detalle en el capítulo 5.1.3.
- b) Captura de puntos:** Objeto de tipo “Static Text”. Sección dentro de la interfaz gráfica que permitirá manejar y controlar el proceso de escaneo, comprende los siguientes botones.
- b.1) Iniciar captura de puntos:** Objeto de tipo “Edit Text”.para ingresar el puerto de comunicación serial.
- b.2) Finalizar captura de puntos:** Menú desplegable, donde se encuentran las diferentes velocidades de transferencia de datos en Baudios. Para efectos de escaneo, se utilizará **11520**.
- b.3) Exportar datos:** Objeto de tipo “Push Button”. Se da click para ejecutar las líneas de código que permitan establecer la comunicación serial.
- c) Puntos capturados:** Objeto de tipo “Static Text”. Dentro de esta sección aparece un cuadro donde se irán visualizando y almacenando los valores numéricos de las coordenadas x_i, y_i, z_i capturadas en el proceso de escaneo.
- d) Datos recibidos:** Objeto de tipo “Static Text”. Dentro de él aparece un recuadro donde se podrá visualizar la trama de datos proveniente de Arduino al momento de realizar el proceso de escaneo, el cual muestra el ángulo de cada uno de los encoder y sus variaciones en tiempo real, separados por los ya explicados delimitadores a,b,c,d,e.
- e) Gráfica de puntos capturados:** Objeto de tipo “Static Text”. Contiene un cuadro de gráfico que permite visualizar en el espacio tridimensional los puntos capturados en el proceso de escaneo.

5.1.2 Establecer conexión robot – PC

El primer paso para poner en funcionamiento el brazo robot consiste en vincular la parte electrónica del robot con el computador mediante un cable de transferencia de datos, que va desde la placa Arduino hasta cualquier puerto serial del computador.

En caso se desconozca el puerto de comunicación seleccionado para la transferencia de datos, se puede averiguar siguiendo la secuencia en el computador:

Inicio-Panel de Control-Hardware y Sonido-Dispositivos e impresoras-Administrador de dispositivos-Puertos (COM y LPT). Una vez conocido el puerto serial de comunicación de datos, se procederá a ejecutar Matlab cargando el archivo que contiene la programación del brazo robot, mostrando la interfaz gráfica.

Como parte de la conexión robot – PC, se deberá ingresar en la sección “COM” de la interfaz gráfica, el puerto donde se establecerá la comunicación serial,

además, se deberá ingresar la velocidad de transmisión de datos, que para el proceso de escaneo y según la programación realizada será siempre 11520 baudios.

5.1.3 Colocación de la posición inicial

Una vez ingresado el puerto de comunicación serial y la velocidad de transferencia de datos en la interfaz gráfica, se deberá colocar el brazo robot en la posición mostrada en la figura 5.2, la cual será llamada “Posición Inicial”, ya que en ella al dar click en establecer conexión todos los encoders incrementales del brazo robot registran un ángulo igual a cero, el cual será usado como referencia para los giros posteriores y el cálculo de las coordenadas respectivas.



Figura 5.2. Posición inicial de referencia.

Fuente: Elaboración propia.

Se da click en “Establecer Conexión” y se espera unos segundos hasta que se inicie correctamente la comunicación serial Arduino - Matlab, luego se da click en “Iniciar Comunicación” para que comience la toma de datos respectiva.

Se puede apreciar que inmediatamente después de dar click en “Iniciar Comunicación”, aparecerá en la sección de “Datos Recibidos”, la trama de valores iniciales que envía Arduino; al inicio éstos son “cero” porque corresponde al ángulo de inicio de cada uno de los encoders incrementales.

5.1.4 Captura de datos

A continuación, se llevará el brazo robot hasta la pieza o curvatura donde se pretenda iniciar el proceso de escaneo y recién en ese punto, se dará click en “Iniciar Captura de Puntos”.

A medida que se va realizando el proceso de escaneo, se irá visualizando en la sección de “Puntos Capturados”, los puntos escaneados y su respectiva localización en el espacio tridimensional tal y como se muestra la figura 5.3.

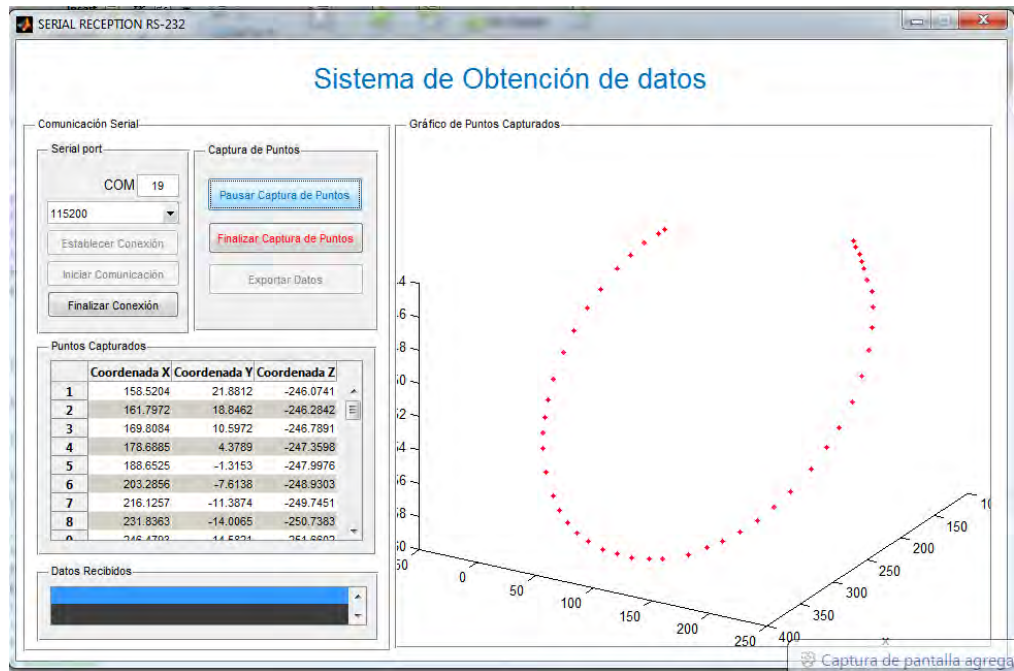


Figura 5.3. Interfaz gráfica en funcionamiento

Fuente: Matlab.

Se puede apreciar además, que mientras se realiza el proceso de escaneo el botón de “Iniciar Captura de Puntos” ha cambiado a “Pausar Captura de Puntos”, esto es para los casos en los que se desee realizar una pausa o en los que no se desee considerar algunos puntos de escaneo, retomando nuevamente la captura dando click en el mismo botón.

Una vez culminado el proceso de escaneo, se da click en “Finalizar Captura de puntos” y luego en “Exportar Datos”. Aparece una ventana como la mostrada en la figura 5.4. la cual permite guardar los puntos x_i, y_i, z_i capturados en un archivo que por defecto será con formato txt.

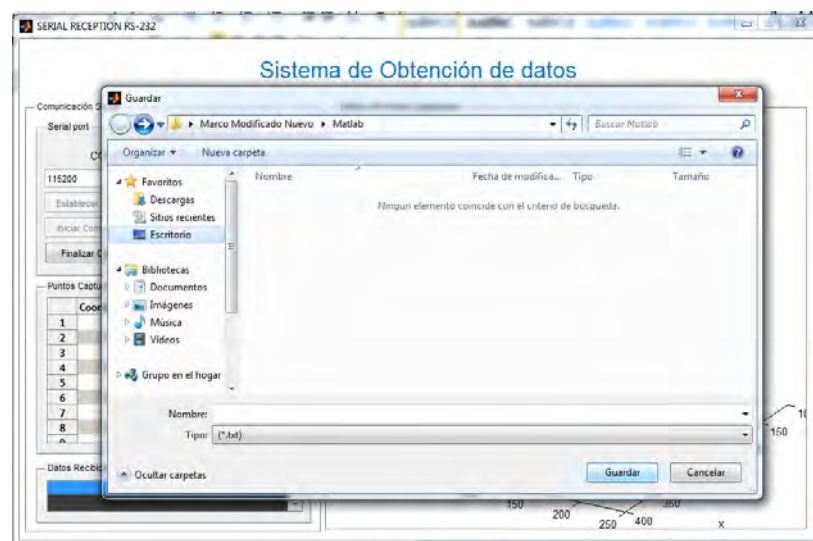


Figura 5.4. Exportar datos de Matlab en formato txt.

Fuente: Matlab

Una vez guardada la data de coordenadas con formato txt, se debe ajustar una curva y se compara la curva con las dimensiones teóricas, para ello se hace uso de Solidworks y su herramienta para el ajuste de curvas dado un conjunto de coordenadas x_i, y_i, z_i .

5.1.5 Ajuste de curva en Solidworks

La figura 5.5 muestra los pasos a seguir para realizar el ajuste de una curva en Solidworks dado un conjunto de coordenadas x_i, y_i, z_i , mediante la siguiente secuencia: Insertar-Curva-Curva por puntos xyz. A continuación, se abrirá la dada de puntos que fue exportada desde Matlab en formato txt, seleccionando después los ejes xyz antes de dar click en “aceptar” según lo muestra las figuras 5.5 y 5.6.

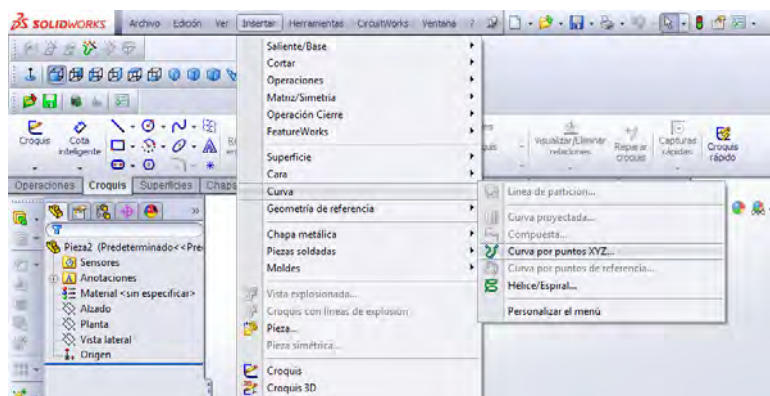


Figura 5.5. Ruta para ajustar curva a una data de puntos.
Fuente: Solidworks

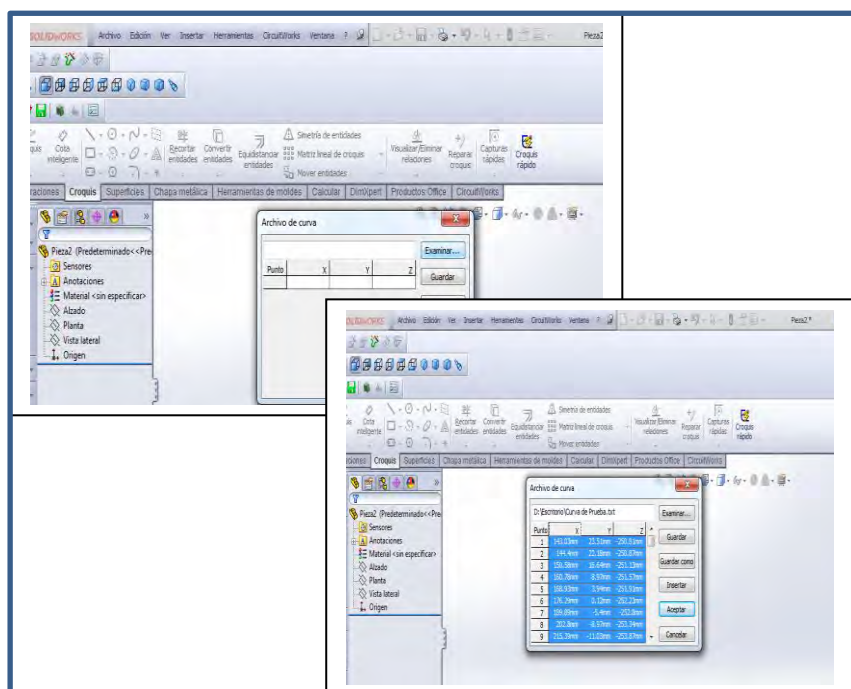


Figura 5.6. Pasos para ajustar curva a una data de puntos
Fuente: Solidworks

Cada uno de los pasos anteriores permiten realizar el proceso de escaneo satisfactoriamente.

5.2 Realización de ensayos

5.2.1 Ensayo de escaneo de curva de prueba

El primer ensayo consiste en calcular las coordenadas en el espacio tridimensional de la curva que resulta al girar solamente el encoder más próximo al pulsador o último encoder. Para ello se debe colocar el brazo robot en una posición similar a la mostrada en la Figura 5.7, de tal manera que sólo se pueda girar el encoder más próximo al pulsador, formando casi completamente una circunferencia.



Figura 5.7. Primera prueba a desarrollar.

Fuente: elaboración propia

Según las dimensiones del brazo, la circunferencia que se formará tendrá teóricamente un radio de 122 mm.

Realizando el ajuste de la curva en Solidworks, se obtendrá la curvatura mostrada en la Figura 5.8.

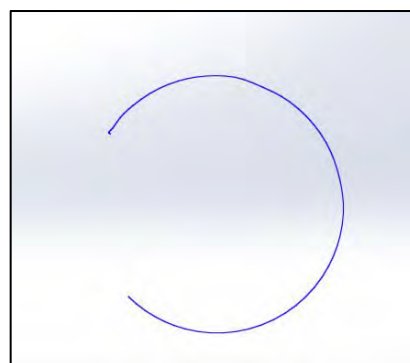


Figura 5.8. Ajuste de una curva por la data de puntos capturados.

Fuente: Solidworks

En la Figura 5.9 se ha superpuesto en la curva de ajuste la circunferencia teórica de 240 mm de diámetro que se forma según las dimensiones del brazo robot.

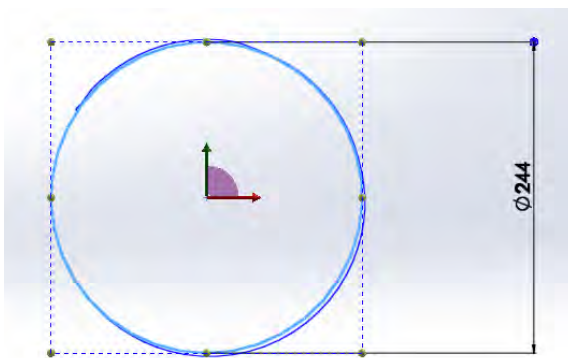


Figura 5.9. Comparación entre la curva de ajuste y la curva teórica de la prueba 1.

Fuente: Solidworks

5.2.1.1 Cálculo de error

Para este primer ensayo, se calcula el error relativo entre el área comprendida de la curva resultado del escaneo y el área comprendida de la curva teórica o real, teniendo como referencia las dimensiones de la figura 5.10.

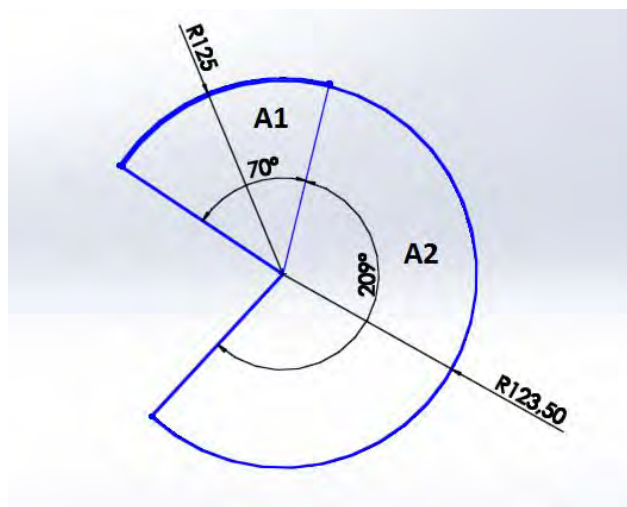


Figura 5.10. Área del Ensayo - curva de prueba.

Fuente: Solidworks.

$$A_1 = \frac{\pi * r^2 * \alpha}{360} = \frac{\pi * (125)^2 * 70}{360} = 9544.77\text{mm}^2$$

$$A_2 = \frac{\pi * r^2 * \alpha}{360} = \frac{\pi * (123.5)^2 * 209}{360} = 27818.11\text{mm}^2$$

Por lo tanto, el área comprendida entre la curva resultado del escaneo resulta igual a:

$$A_{\text{TOTAL}} = A_1 + A_2 = 37362.88\text{mm}^2$$

A continuación, conocidas las dimensiones de la curva real que tiene como radio 122 mm, se calcula el área teórica.

$$A_{\text{TEÓRICA}} = \frac{\pi * r^2 * \alpha}{360} = \frac{\pi * (122)^2 * 279}{360} = 36238.58 \text{mm}^2$$

Así, el error relativo de este primer ensayo tomando como referencia las área bajo ambas curvas, resulta igual a 3.1 %.

$$\begin{aligned} \text{Error}_{\text{Relativo}} &= \frac{A_{\text{MEDIDA}} - A_{\text{REAL}}}{A_{\text{REAL}}} * 100 \\ &= \frac{37362.88 - 36238.58}{36238.58} * 100 \end{aligned}$$

$$\text{Error}_{\text{Relativo}} = +3.1\%$$

5.2.2 Ensayo de escaneo de una circunferencia

La segunda prueba busca escanear la circunferencia externa de un disco de 122 mm de diámetro, para este caso a diferencia del primer ensayo, se visualizará el funcionamiento de los cinco encoder en conjunto (figura 5.11).



Figura 5.11. Escaneo de la circunferencia externa de un disco.

Fuente: elaboración propia.

El ajuste de la nueva curva en Solidworks, es la mostrada en la Figura 5.12.

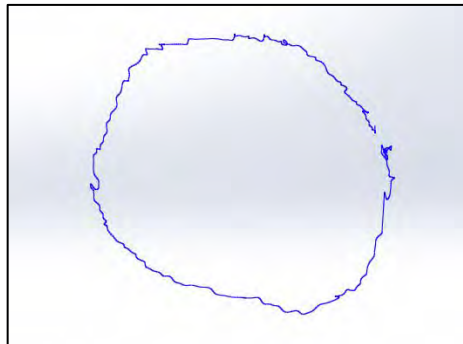


Figura 5.12. Ajuste de una curva por los puntos capturados en la prueba 2.

Fuente: Solidworks.

En la Figura 5.13 se muestra la circunferencia real de 122 mm de diámetro sobre dibujado en la curva de ajuste inicial, se puede apreciar claramente un mayor error dado que los cinco encoder trabajan ahora en conjunto y el procesamiento de datos es más complejo.

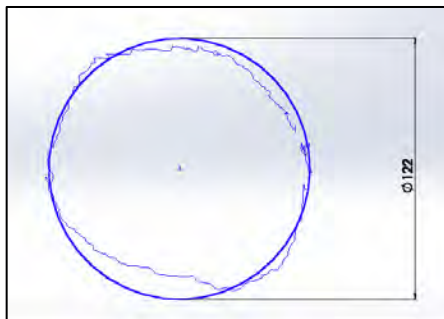


Figura 5.13. Comparación de la curva teórica con la curva de ajuste de la prueba 2.

Fuente: elaboración propia.

5.2.2.1 Cálculo de error

Para el cálculo del área bajo la curva resultado del escaneo, se realizará mediante la sumatoria de áreas de triángulos tal y como se muestra en la Figura 5.14, cuyas dimensiones fueron obtenidas del programa Solidworks.

$$A_{\text{TOTAL}} = \sum_{i=1}^{17} A_i = 10397.313 \text{ mm}^2$$

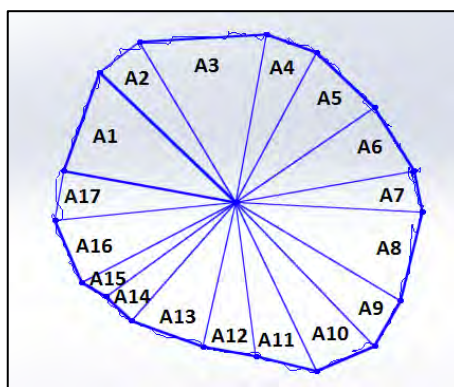


Figura 5.14. Área del ensayo – escaneo de una circunferencia.

Fuente: elaboración propia.

Conocidas las dimensiones de la curva real que tiene como radio 61 mm, se calcula el área teórica.

$$A_{\text{TEÓRICA}} = \pi * r^2 = \pi * (61)^2 = 11689.87 \text{ mm}^2$$

Así, el error relativo del segundo ensayo resulta igual a -11.05 %.

$$\text{Error}_{\text{Relativo}} = \frac{A_{\text{MEDIDA}} - A_{\text{REAL}}}{A_{\text{REAL}}} * 100$$

$$\text{Error}_{\text{Relativo}} = \frac{10397.313 - 11689.87}{11689.87} * 100$$

$$\text{Error}_{\text{Relativo}} = -11.05\%$$

5.2.3 Ensayo de escaneo del perímetro de una pieza rectangular

El tercer ensayo consiste en escanear el perímetro de una pieza rectangular de 120 mm x 62 mm. Al igual que la prueba anterior, se visualizará nuevamente un trabajo simultáneo de todos los encoder.



Figura 5.15. Escaneo del perímetro de una pieza rectangular.

Fuente: elaboración propia.

Una vez guardada la data de coordenadas, se procederá a realizar el ajuste de la curva en Solidworks, la cual se muestra en la figura 5.16.}.



Figura 5.16. Ajuste de una curva por los puntos capturados en la prueba 3.

Fuente: elaboración propia.

La figura 5.17 muestra la curva rectangular teórica de 120 mm x 62 mm así como la curva que realizó el ajuste por los puntos obtenidos de la captura de datos. Como se puede apreciar, el error es mucho menor comparado con el de la prueba 2, esto es debido a que para esta nueva prueba el trabajo en conjunto de los cinco encoder simultáneamente es menos influyente ya que el movimiento es prácticamente lineal.

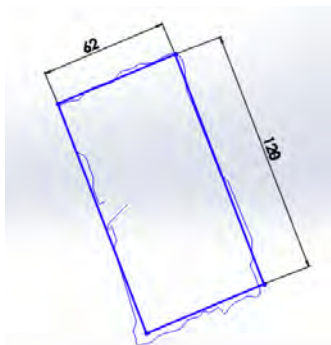


Figura 5.17. Comparación de la curva teórica con la curva de ajuste de la prueba 3.

Fuente: elaboración propia.

5.2.3.1. Cálculo de error

Para este nuevo cálculo, el área bajo la curva resultado del escaneo, se realizará mediante la sumatoria de dos áreas de cuadriláteros tal y como se muestra en la Figura 5.18, cuyas dimensiones fueron obtenidas del programa Solidworks.

Por lo tanto, el área resulta igual a:

$$A_{TOTAL} = A_1 + A_2 = 7675.741mm^2$$

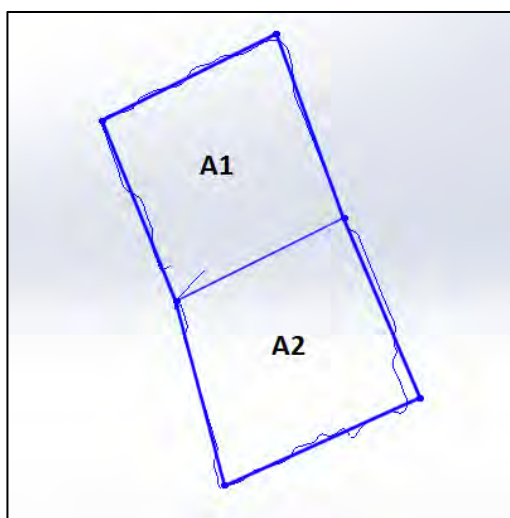


Figura 5.18. Área del ensayo – escaneo del perímetro de una pieza rectangular.

Fuente: Elaboración propia.

Conocidas las dimensiones de la curva rectangular teórica o real cuyos lados son de 62 mm x 120 mm, su área será igual a:

$$A_{TEÓRICA} = 62 \times 120 = 7440mm^2$$

Así, el error relativo del segundo ensayo resulta igual a +3.17 %.

$$Error_{Relativo} = \frac{A_{MEDIDA} - A_{REAL}}{A_{REAL}} * 100 = \frac{7675.741 - 7440}{7440} * 100$$

$$Error_{Relativo} = +3.17\%$$

Conclusiones

- Para desarrollar el brazo robótico se partió de una estructura básica con cinco eslabones y cinco grados de libertad porque se asemeja a la anatomía humana y proporciona los movimientos necesarios para la realización del escaneo de curvas.
- El análisis estático de la estructura básica permitió determinar el peso mínimo de la base para evitar el fenómeno de volcadura, éste fue igual a 7.9 kg.
- Para pasar de un brazo mecánico a uno robótico se incorporan transductores o sensores de posición, escogiendo para ello los encoders incrementales, instalados en cada una de las articulaciones.
- El adecuado diseño de cada articulación minimizó los esfuerzos en el eje del encoder debido a las cargas que originan esfuerzos axiales, cortantes y momentos flectores.
- Este trabajo de tesis también permite visualizar las ventajas del uso de las matrices de transformación homogénea por su capacidad de representación conjunta de posición y orientación. Se observa, que sólo basta multiplicar en el orden adecuado las matrices de transformación correspondientes.
- Se demuestra que si se siguen uno a uno los pasos del algoritmo de Denavit – Hartenberg y se escoge adecuadamente los sistemas de coordenadas asociados a cada eslabón, entonces es posible encontrar una relación entre dos eslabones contiguos que permitan encontrar una relación entre dos eslabones no contiguos, haciendo uso de las cuatro transformaciones básicas de D-H (rotaciones y traslaciones), que dependen exclusivamente de las características geométricas del eslabón.
- Programar en Arduino para obtener datos de posición de los transductores es muy sencillo, pero necesita de un software tipo Matlab para desarrollar las matrices de transformación y de una interfaz gráfica para poder visualizar los datos y crear una comunicación entre el programador y el programa.
- Los resultados obtenidos en los diferentes ensayos realizados al superponer las curvas teóricas sobre las curvas escaneadas, demuestran que al trabajar con un menor número de encoders en simultáneo, el proceso de escaneo es mucho más eficiente; porque la programación en Arduino está diseñada para preguntar secuencialmente por el cambio de cada uno de los encoders hasta que termine el bucle. Un cambio en paralelo de otro encoder puede pasar desapercibido hasta que se genere un nuevo bucle, generando errores de posición.
- Se ha cumplido exitosamente con los objetivos de la presente tesis ya que el brazo robot desarrollado, es un prototipo que presenta un diseño mecánico eficaz y un desarrollo electrónico sencillo usando herramientas de fácil acceso para los usuarios, permitiendo una captura de información aceptable para su posterior procesamiento.

Recomendaciones

- Según los resultados obtenidos en el capítulo 4, se ha demostrado que el código de programación en Arduino para la captura de datos de los encoder, resulta eficiente para determinadas condiciones de operación, sin embargo, cuando se desea escanear piezas mucho más complicadas, se vuelve ineficiente.
- Para una mejor captura de datos en tiempo real, se recomienda trabajar el procesamiento de información con un Arduino de mayor capacidad para cada encoder. Otra alternativa resultaría de trabajar con herramientas de captura de información más sofisticadas como “Psoc” o utilizando sistemas automáticos de control con tiempos de muestreo.

Bibliografía

- Balcells J. y Romeral J. (1997). Autómatas programables. Grafiques Torreasa. Barcelona. España.
- Barrientos A., Peñin L., Balaguer C. y Aracil R. (2007). Fundamentos de robótica. Universidad Politécnica de Madrid.
- Boltom W. (1995) Mediciones y pruebas eléctricas y electrónicas: Instrumentos automáticos.
- Díaz L. (2005). Análisis y Planeamiento. San José de Costa Rica.
- Erdman A., Sandor G. (1998). Diseño de mecanismos: Análisis y síntesis. Ed. Prentice Hall. México.
- Ferraté et al. (1986). Configuraciones básicas de robots manipuladores industriales.
- International Federation of Robotics (IFR) recuperado de www.ifr.org.
- Knasel T. (1986). Mobile Robots-state of the art review. Elsevier Journal. Vol 2. N° 2, pp 149-155.
- Madrid E., Tafur J., Robles S. y Curay J. (2007). Octavo Congreso Ibero americano de Ingeniería Mecánica. Cusco.
- Martínez G., Jáquez S., Rivera J., Sandoval R. (2008). Diseño y construcción de un brazo robótico de 5 GDL. Revista de Ingeniería Eléctrica, Electrónica y Computación, Vol 4 N° 1.
- Ojeda C. (2015). Apuntes de mecánica-dinámica. Universidad de Piura.
- Ollero A. (2001). Robótica: manipuladores y robots móviles. Marcombo Editores. Barcelona. España.
- Siciliano B., Sciavicco L., Villani L., Oriolo G. (2009). Robotics, modelling, planning and control. Ed. Springer. London.

Webgrafía

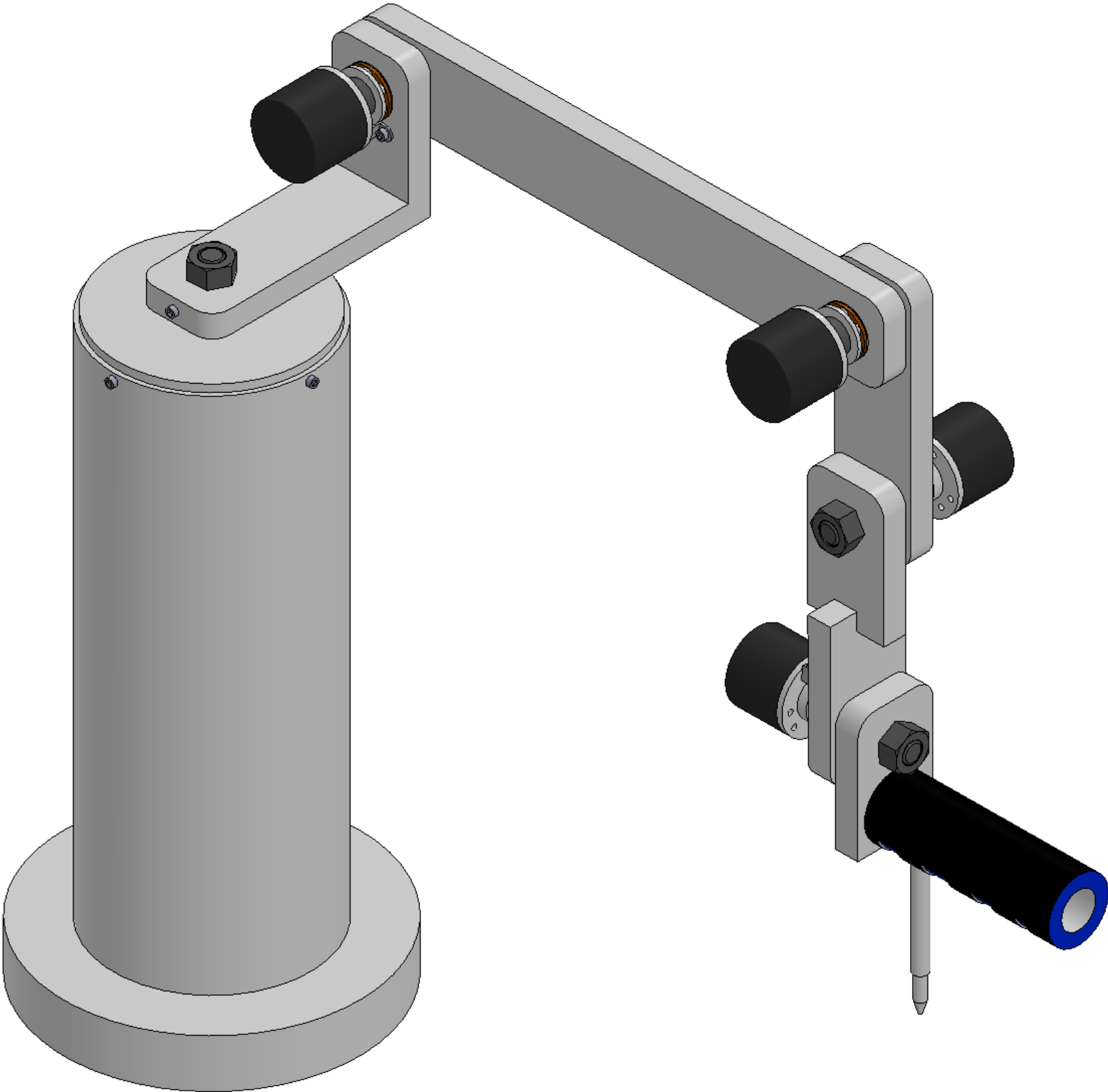
- Abdel-Malek K., Othman S., Erratum to: Multiple sweeping using the Denavit-Hartenberg representation method, 1999, 855. Recuperado de <https://www.deepdyve.com/lp/elsevier/erratum-to-multiple-sweeping-using-the-denavit-hartenberg-eWUzzmznH2>
- Abdel-Malek K., Othman S., Multiple sweeping using the Denavit-Hartenberg representation method, 1999, 567-583. Recuperado de <http://www.sciencedirect.com/science/article/pii/S0010448599000536>
- Arduino. Recuperado de <https://www.arduino.cc/en/Main/ArduinoBoardNano>
- Barrientos A., Álvarez M., Hernández J., Del Cerro J., Rossi C., Modelado de Cadenas Cinemáticas mediante Matrices de Desplazamiento. Una alternativa al método de Denavit-Hartenberg, 2012, 371-382. Recuperado de <http://www.sciencedirect.com/science/article/pii/S1697791212000623>

- Basile F., Caccavale F., Chiacchio P., Coppola J., Curatella C., Task-oriented motion planning for multi-arm robotic systems, 2012, 569-582. Recuperado de <http://www.sciencedirect.com/science/article/pii/S0736584512000191>
- Cortés J., La representación Denavit-Hartenberg, marzo 2008. Recuperado de http://personal.us.es/jcortes/Material/Material_archivos/Articulos%20PDF/RepresentDH.pdf
- Escáner 3D. (2015, 20 de diciembre). Recuperado 10:30, Noviembre 28, 2015, Enciclopedia libre. https://es.wikipedia.org/wiki/Esc%C3%A1ner_3D
- Franco R., (2014, 9 de Diciembre). GUI-Interfaz Gráfica de Usuario en MATLAB. Videos recuperados de <https://www.youtube.com/watch?v=jBecrEnkSV0>
- Gonzáles J., (2013, 16 de Noviembre). ROBÓTICA. Videos recuperados de <https://www.youtube.com/playlist?list=PLik-Tt9YjWT28dqp9itZi7MpYpbGHep2m>
<http://echebarriasuministros.com/images/catalogo/13-tornilleria-y-remaches.pdf>.
- Mineo C., Gareth S., Ian P., Cooper I., Robotic path planning for non-destructive testing-A custom MATLAB toolbox approach. 2016, 1-12. Recuperado de <http://www.sciencedirect.com/science/article/pii/S0736584515000666>
- Paredes J., (2013, 18 de abril). Historia del scanner. Web recuperada de <https://prezi.com/dtlz77ad9ksh/historia-del-scanner/>
- Rocha C., Tonetto C., Dias A., A comparison between the Denavit-Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators, 2011, 723-728. Recuperado de <http://www.sciencedirect.com/science/article/pii/S073658451100010X>
- Sharma A., Mithra M., Design of a low-cost five-inger anthropomorphic robotic arm with nine degrees of freedom, 2012, 551-558. Recuperado de <http://www.sciencedirect.com/science/article/pii/S0736584512000026>
- Steinparz F., Co-ordinate transformation and robot control with Denavit-Hartenberg Matrices, 1985, 8, 303-316. Recuperado de <http://www.sciencedirect.com/science/article/pii/0745713885900314>
- Tornillería y remaches. Recuperado de
- Zhou L., Bai S., Rygaard M., Integrated dimensional and drive-train design optimization of a light-weight anthropomorphic arm, 2012, 113-122. Recuperado de <http://www.sciencedirect.com/science/article/pii/S0921889011001771>
- Zunt D. (2004). Who did actually invent the word "robot" and what does it mean? Recuperado de <http://web.archive.org/web/20150415062618/http://capek.misto.cz/english/robot.html>.

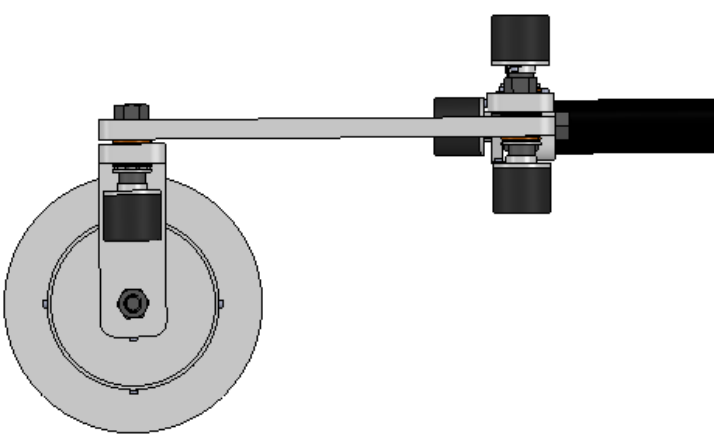
ANEXOS

ANEXO A : PLANOS

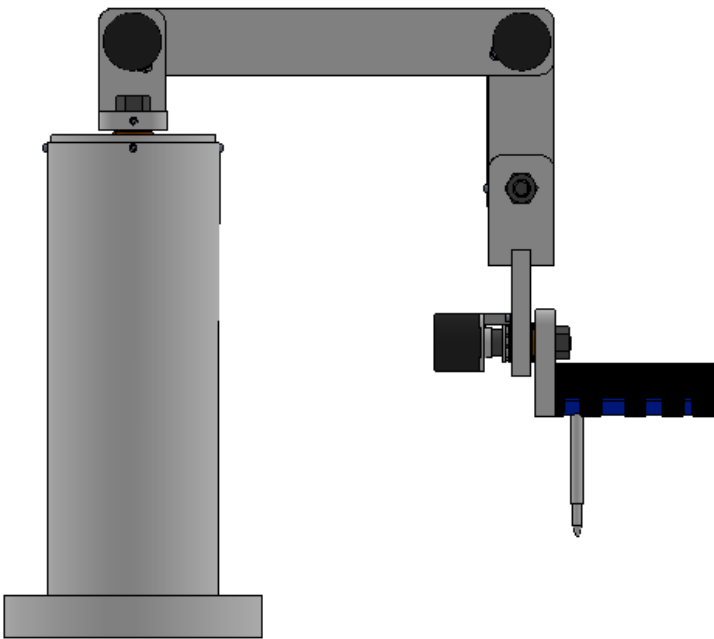
VISTA ISOMÉTRICA
ESCALA 1:2



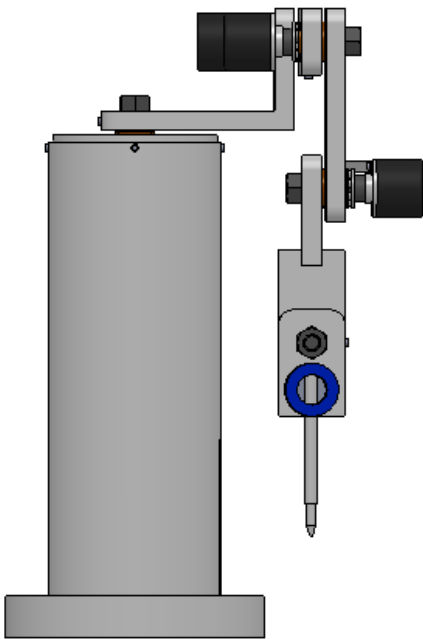
VISTA HORIZONTAL
ESCALA 1:5



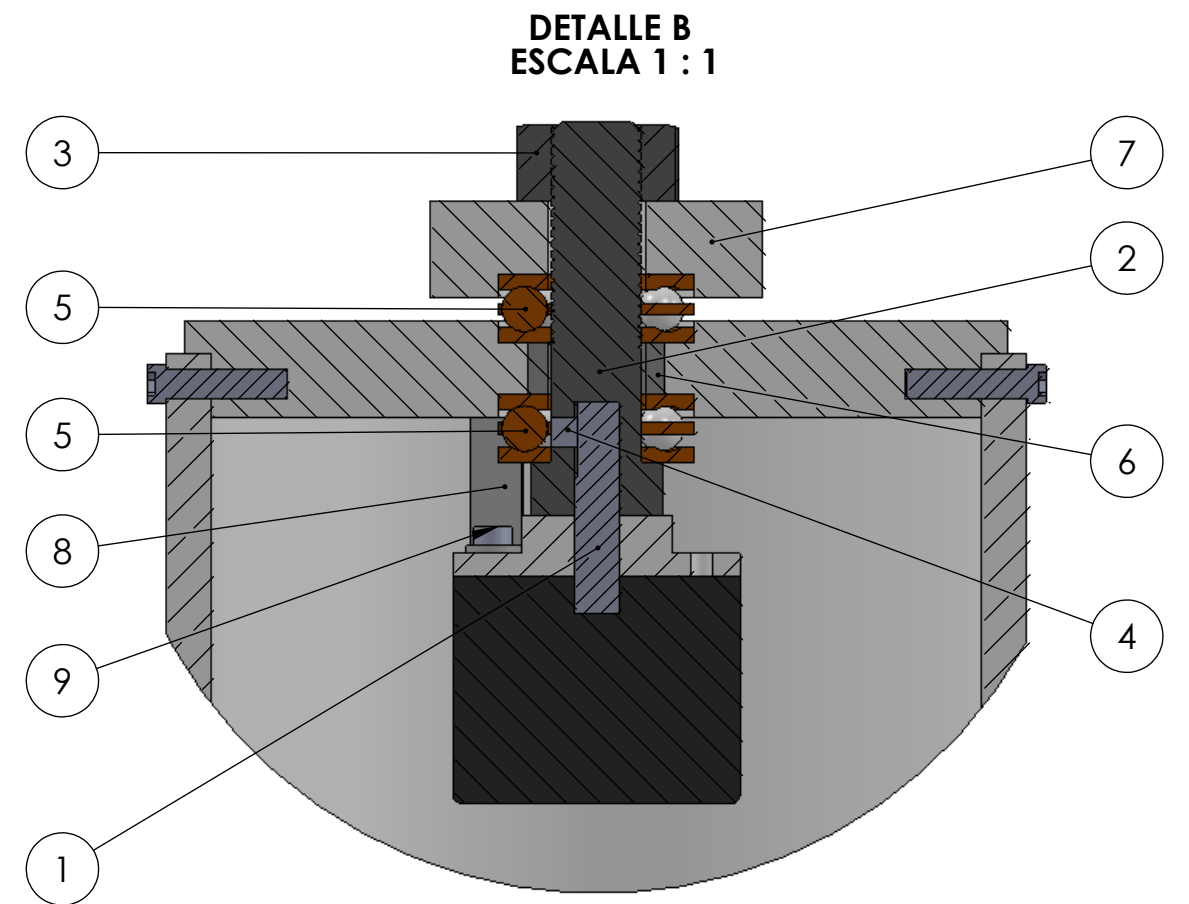
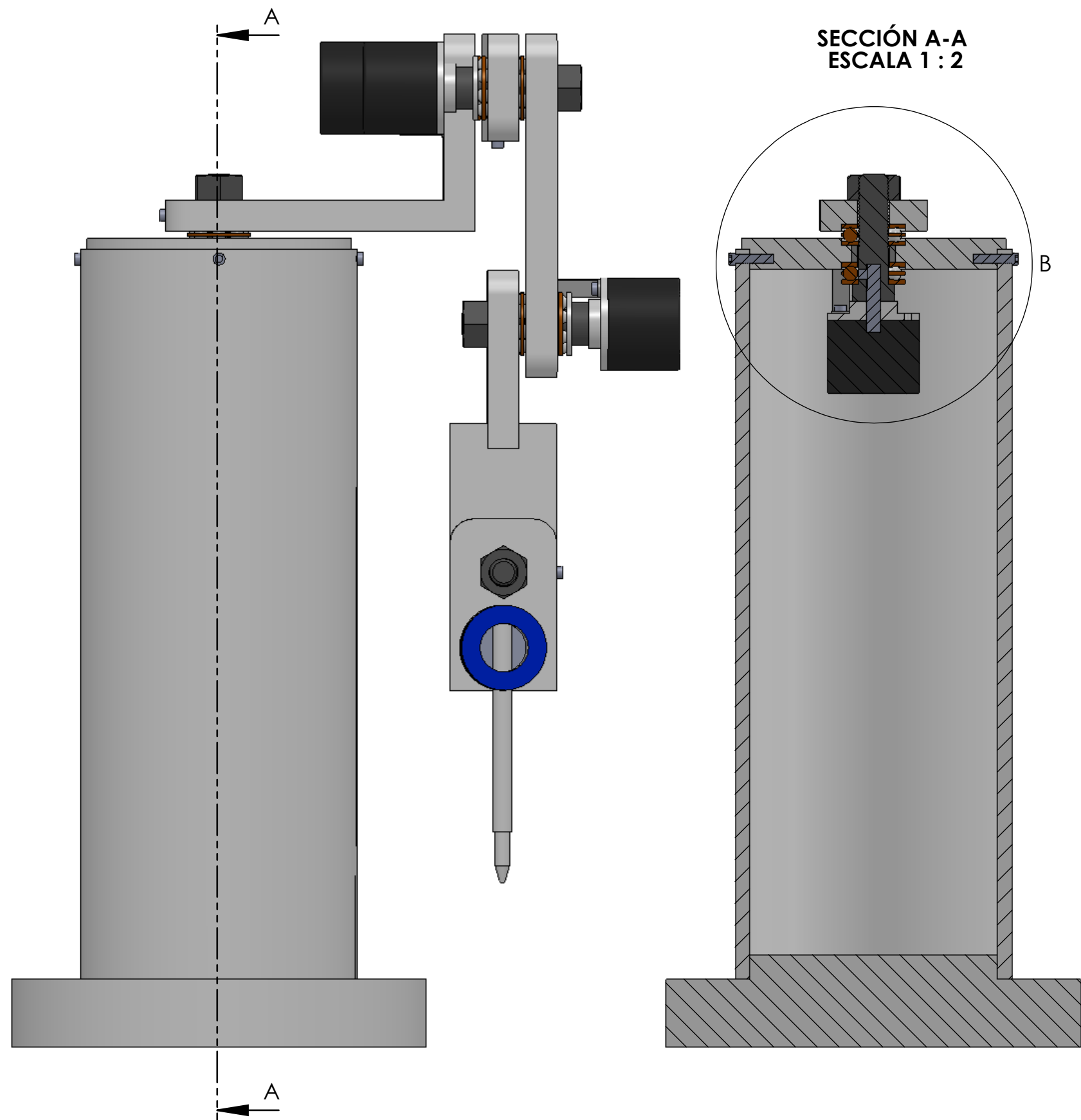
VISTA FRONTAL
ESCALA 1:5



VISTA PERFIL
ESCALA 1:5



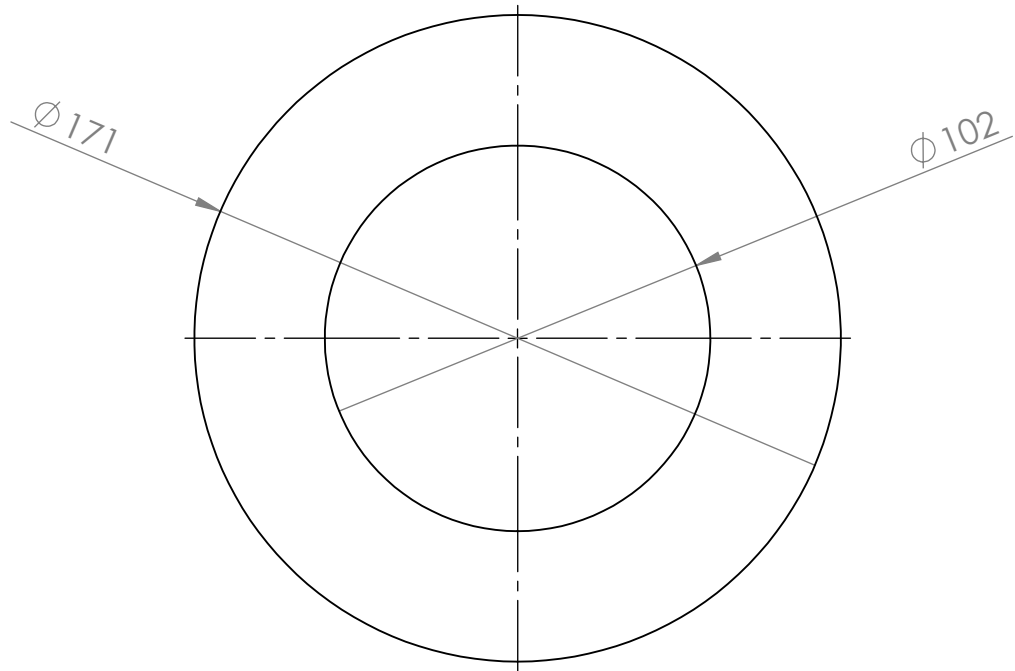
	FECHA	NOMBRE	UNIVERSIDAD DE PIURA FACULTAD DE INGENIERÍA	
DIBUJADO	01/10/16	MATR		
REVISADO	01/10/16	MATR		
TÍTULO			ENSAMBLAJE	ESCALA
ANEXO A-1				1:2
NOMBRE Y APELLIDO				PLANO
MARCO AURELIO TRONCOS RIOFRIO				Nº 01



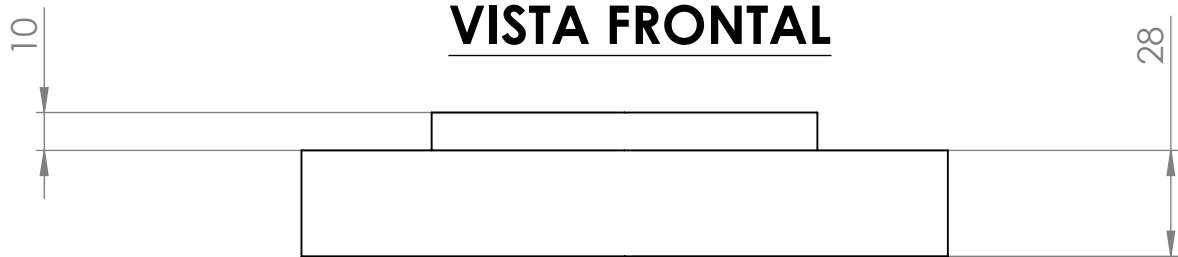
N.º DE ELEMENTO	N.º DE PIEZA	CANTIDAD TOTAL
1	Eje Encoder	5
2	Perno M12x40	5
3	Tuerca M12	7
4	Prisionero	5
5	Rodamiento Axial	10
6	Bocina	5
7	Brazo 1	1
8	Unión	5
9	Perno M3x3.5	5

	FECHA	NOMBRE	UNIVERSIDAD DE PIURA FACULTAD DE INGENIERÍA	
DIBUJADO	01/10/16	MATR		
REVISADO	01/10/16	MATR		
TÍTULO			VISTA SECCIÓN	ESCALA
ANEXO A-2				1:1
NOMBRE Y APELLIDO			VISTA SECCIÓN	PLANO
MARCO AURELIO TRONCOS RIOFRIO				Nº 02

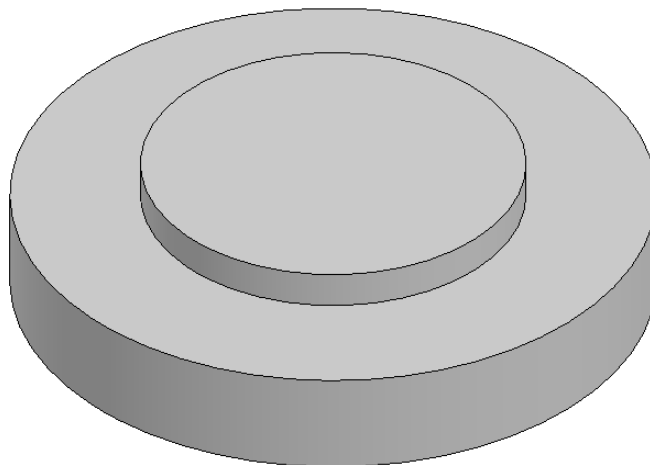
VISTA HORIZONTAL



VISTA FRONTAL



VISTA ISOMÉTRICA



	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-3

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

DISCO BASE

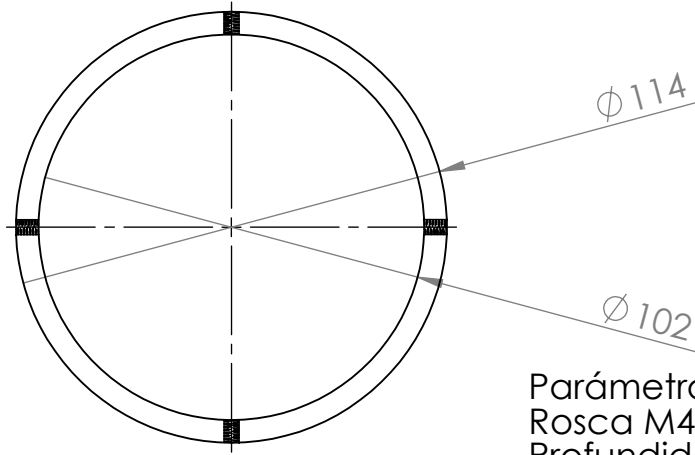
ESCALA

1:2

PLANO

N° 03

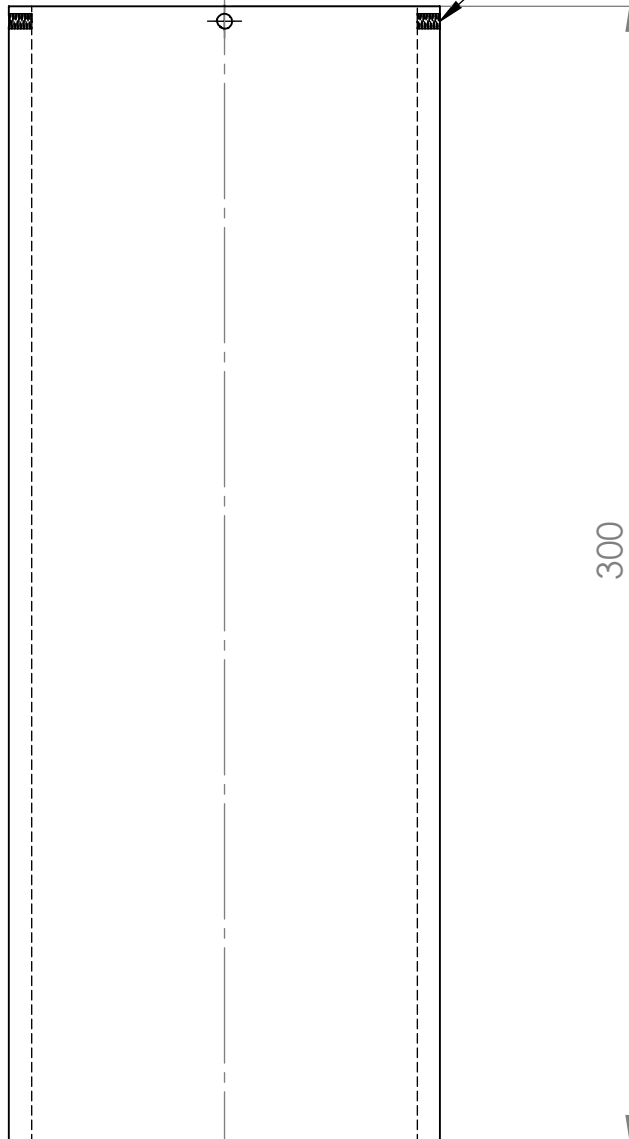
VISTA HORIZONTAL



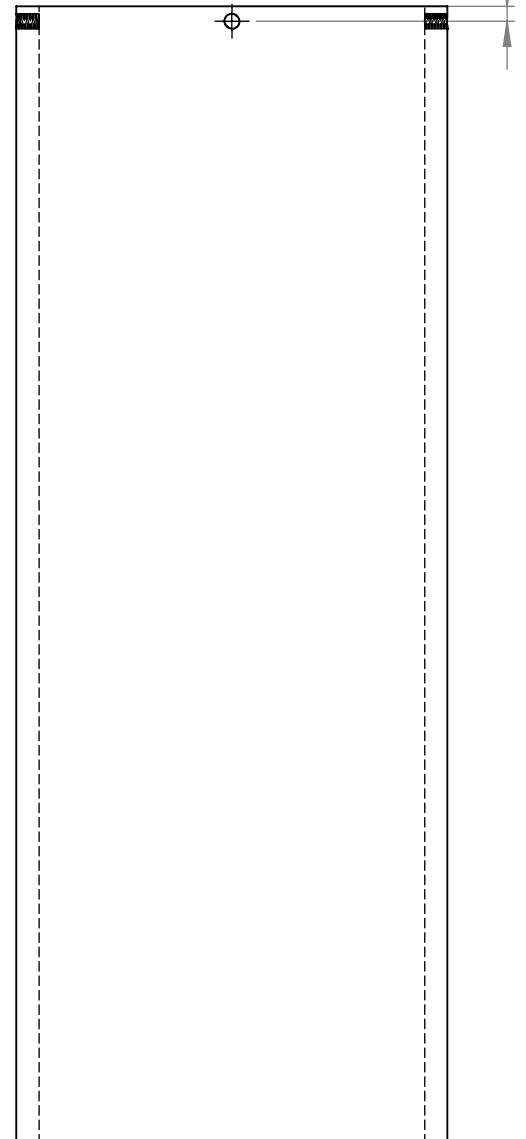
VISTA ISOMÉTRICA



VISTA FRONTAL



VISTA PERFIL



Parámetro de Rosca
Rosca M4
Profundidad: 6mm

	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-4

CILINDRO BASE

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

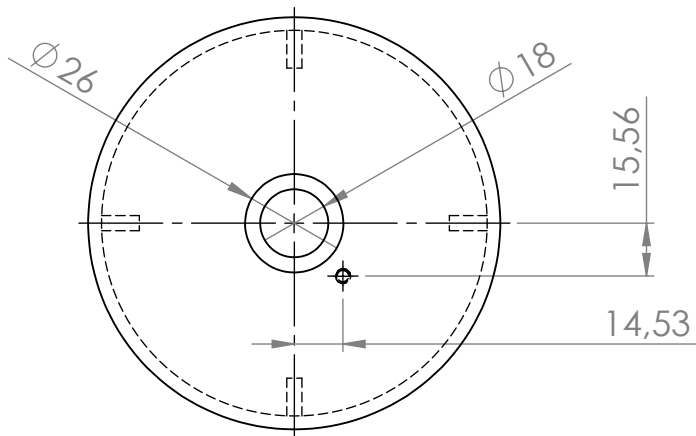
ESCALA

1:2

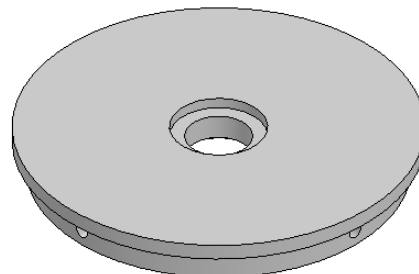
PLANO

Nº 04

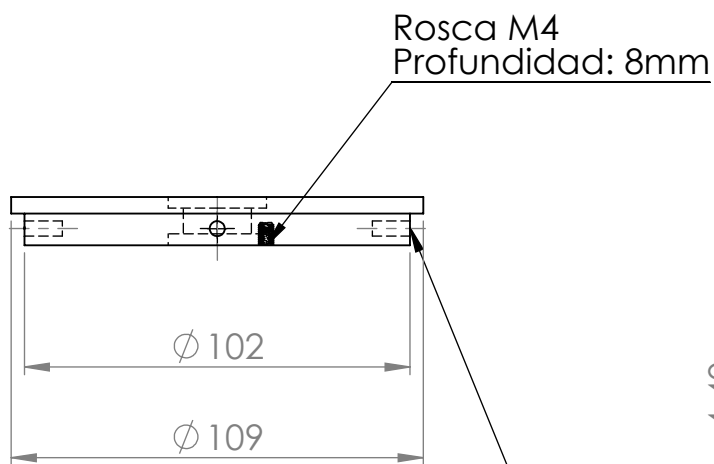
VISTA HORIZONTAL



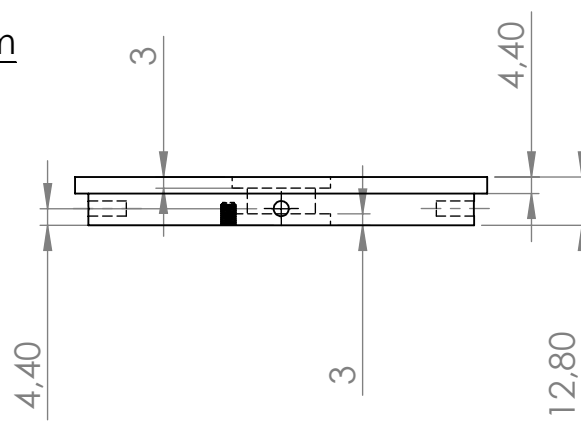
VISTA ISOMÉTRICA



VISTA FRONTAL



VISTA PERFIL



Parámetro de Agujeros
Profundidad: 10 mm

	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-5

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

TAPA BASE

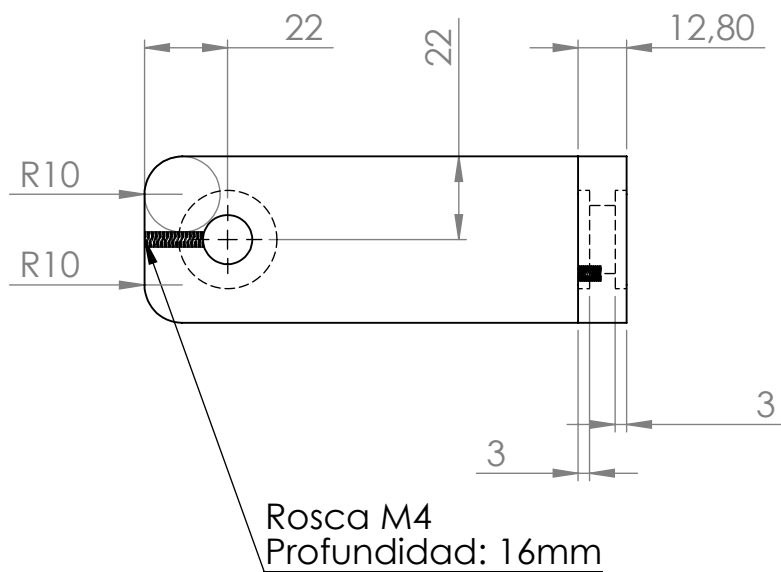
ESCALA

1:2

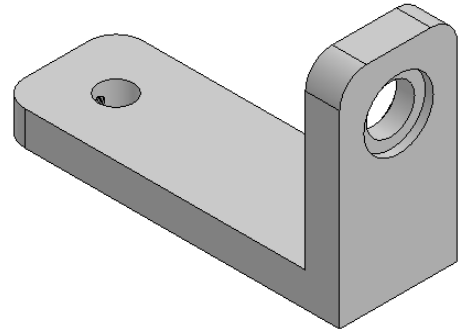
PLANO

N° 05

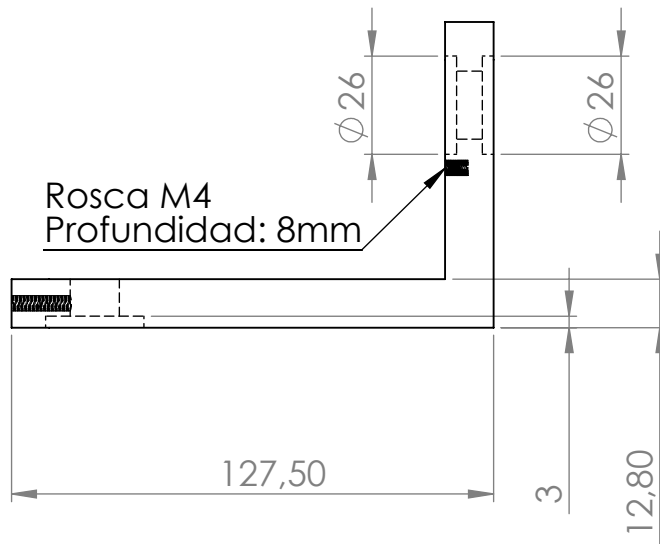
VISTA HORIZONTAL



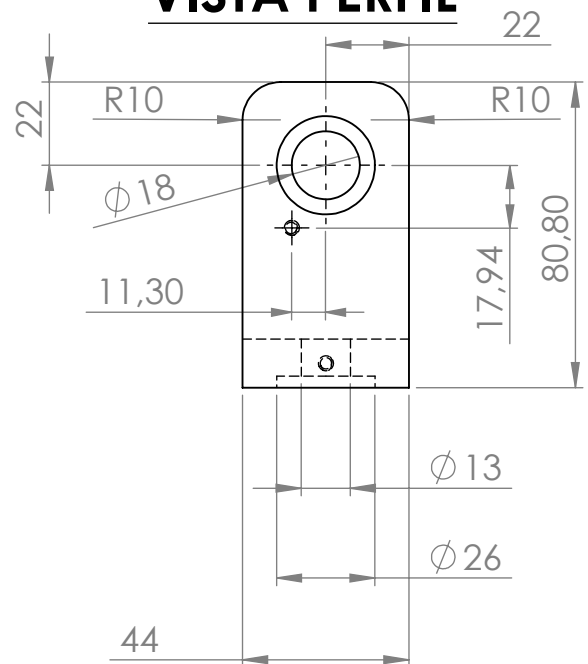
VISTA ISOMÉTRICA



VISTA FRONTAL



VISTA PERFIL



	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-6

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

**CUERPO DEL
BRAZO ROBOT**

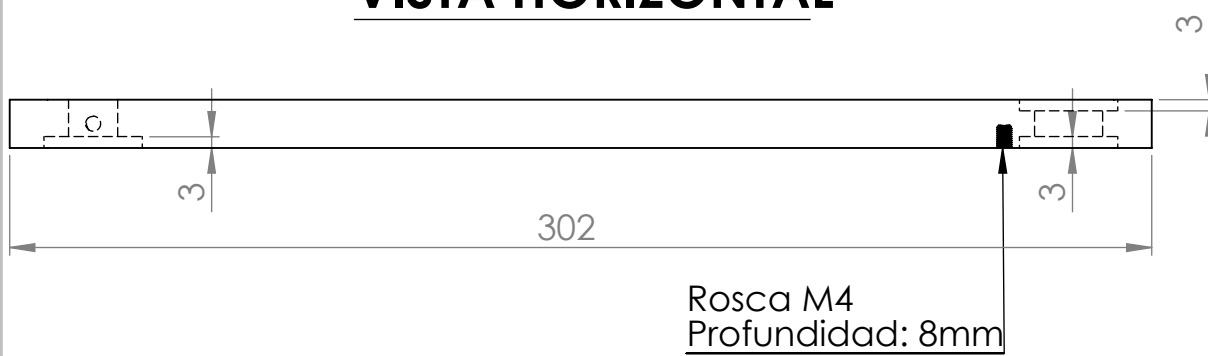
ESCALA

1:2

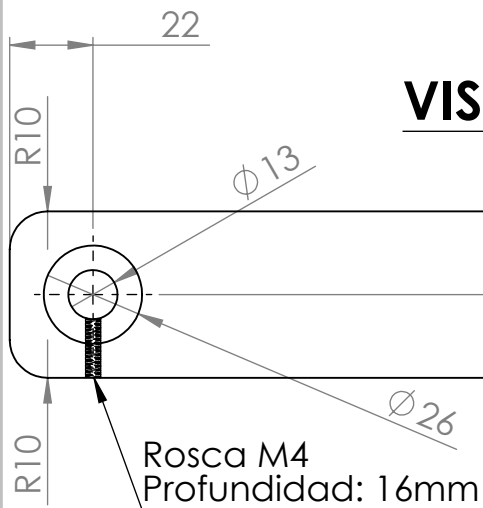
PLANO

N° 06

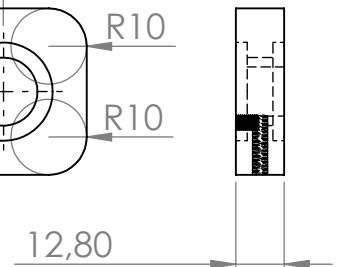
VISTA HORIZONTAL



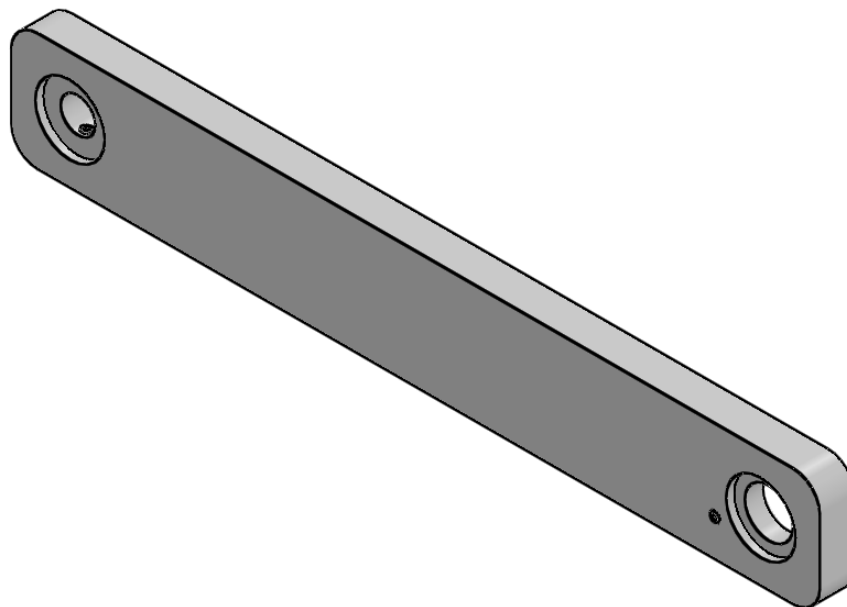
VISTA FRONTAL



VISTA PERFIL



VISTA ISOMÉTRICA



	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-7

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

BRAZO

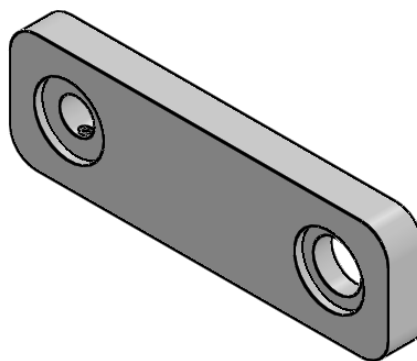
ESCALA

1:2

PLANO

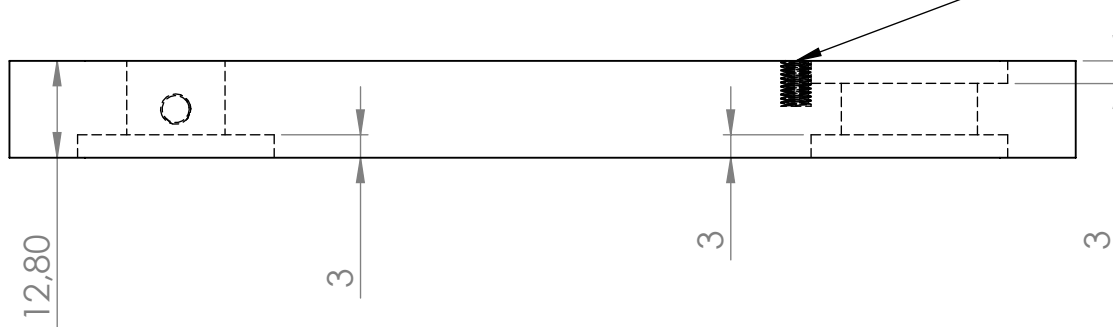
N° 07

VISTA ISOMÉTRICA



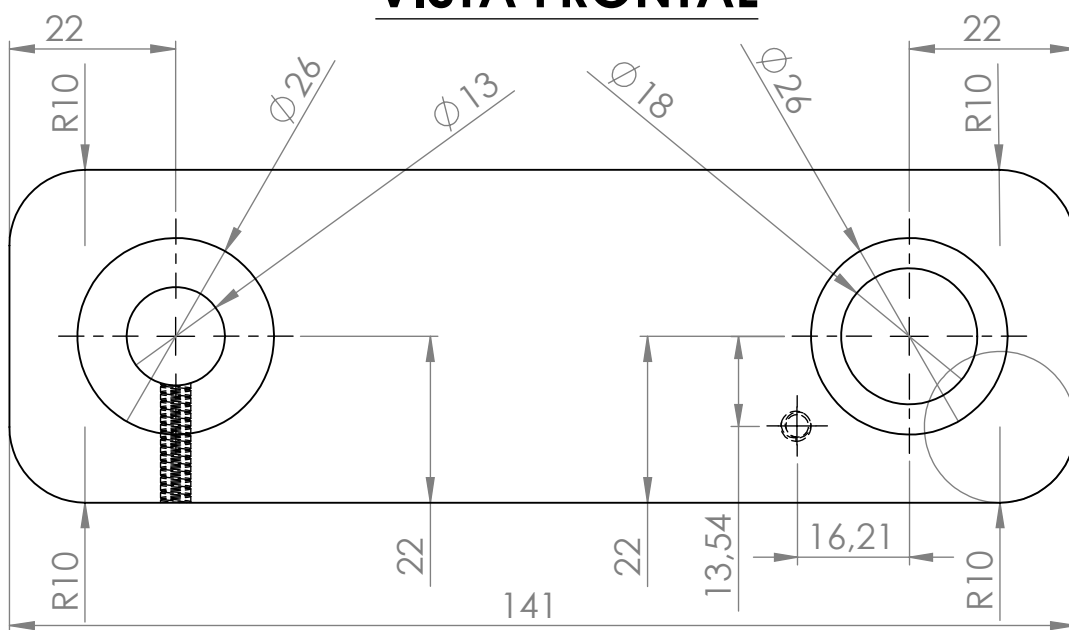
VISTA HORIZONTAL

Rosca M4
Profundidad: 8mm



VISTA FRONTAL

VISTA PERFIL



Rosca M4
Profundidad: 16 mm

	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-8

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

ANTEBRAZO

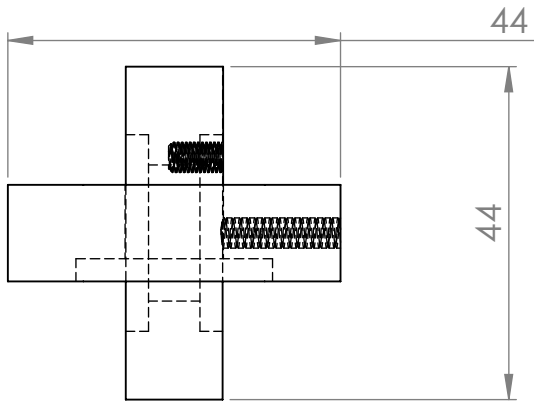
ESCALA

1:1

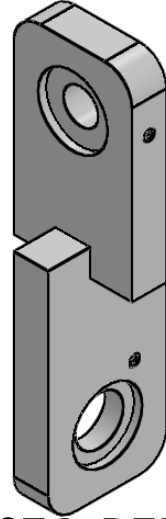
PLANO

N° 08

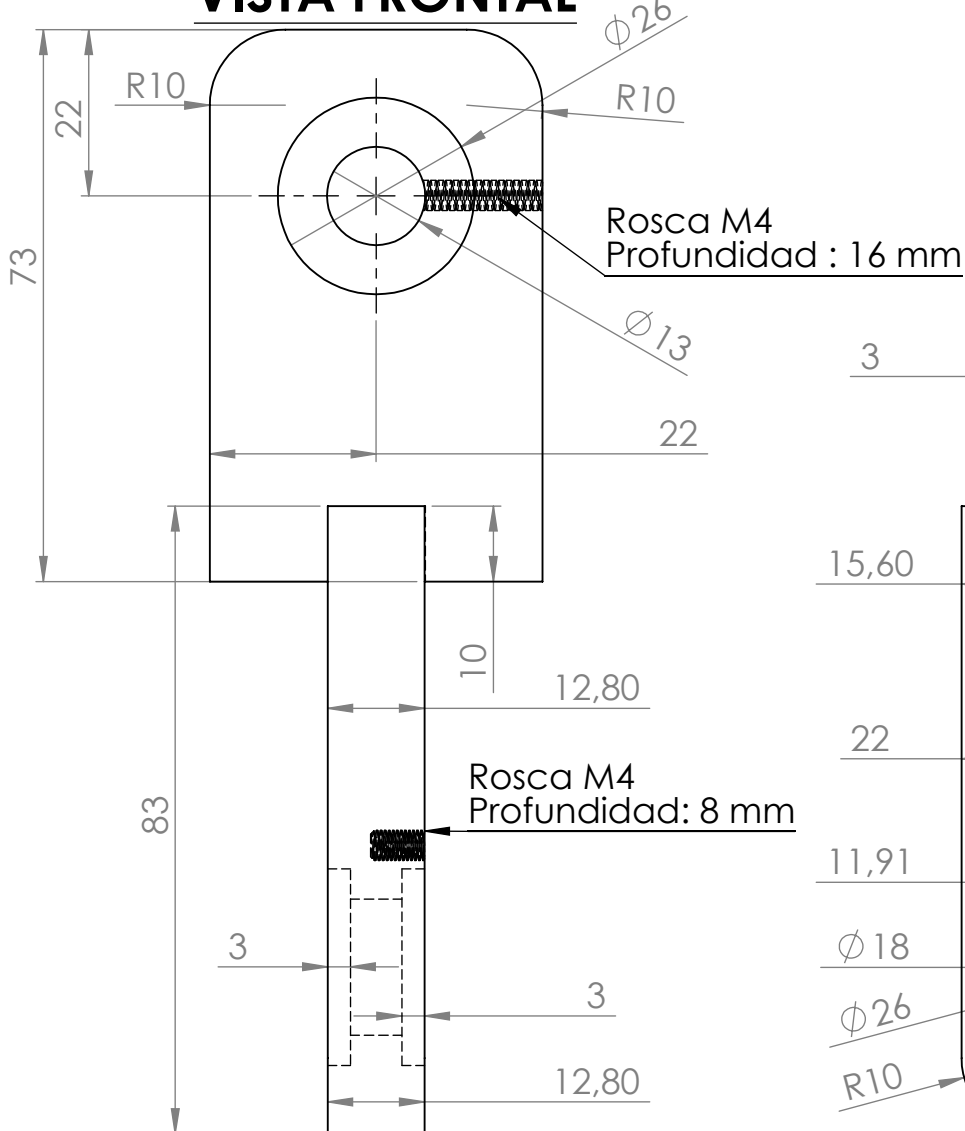
VISTA HORIZONTAL



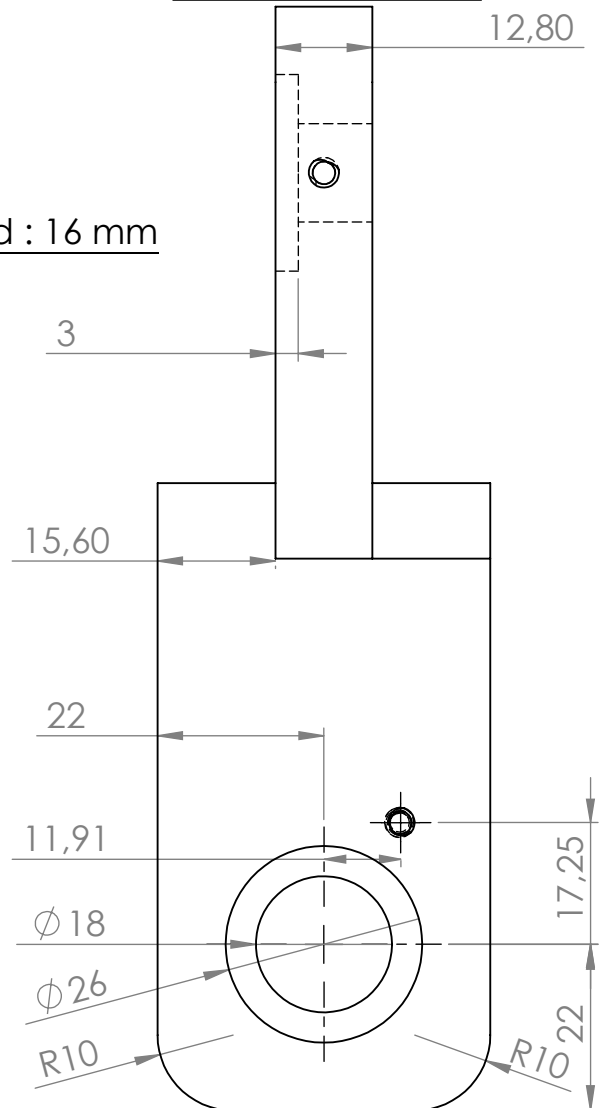
VISTA ISOMÉTRICA



VISTA FRONTAL



VISTA PERFIL



	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-9

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

MANO

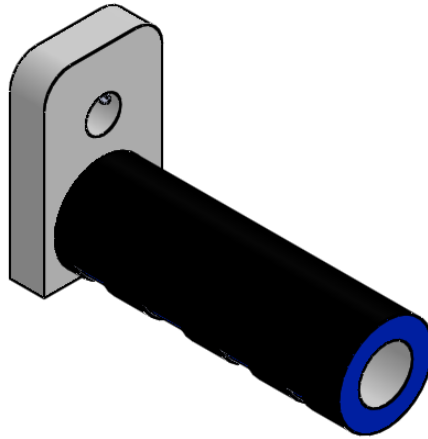
ESCALA

1:1

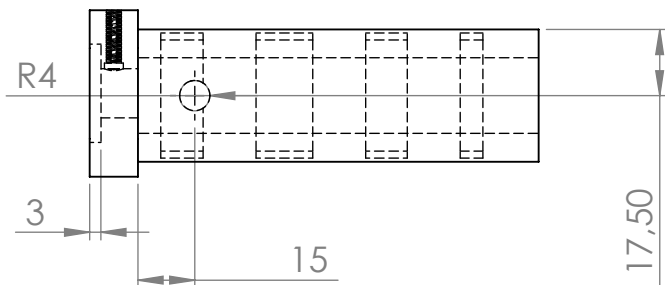
PLANO

N° 09

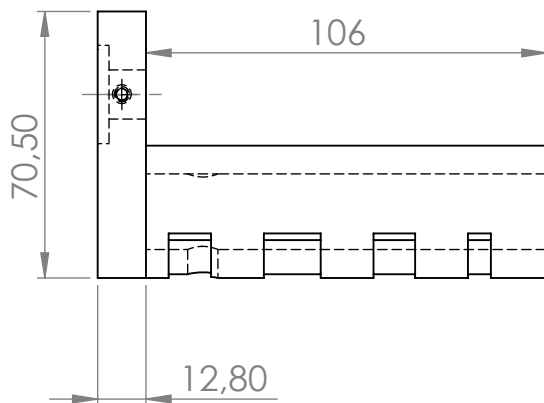
VISTA ISOMÉTRICA



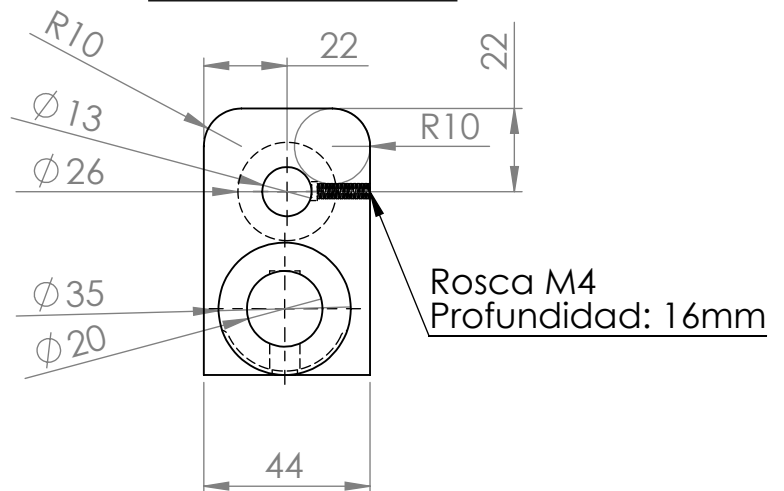
VISTA HORIZONTAL



VISTA FRONTAL



VISTA PERFIL



	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-10

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

**COMPONENTE
FINAL**

ESCALA

1:2

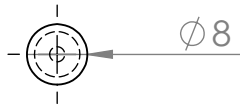
PLANO

N° 10

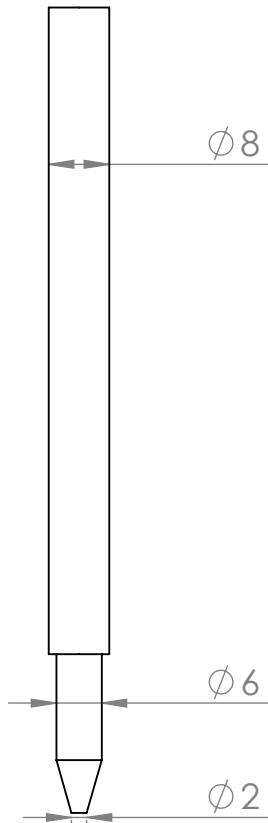
VISTA ISOMÉTRICA



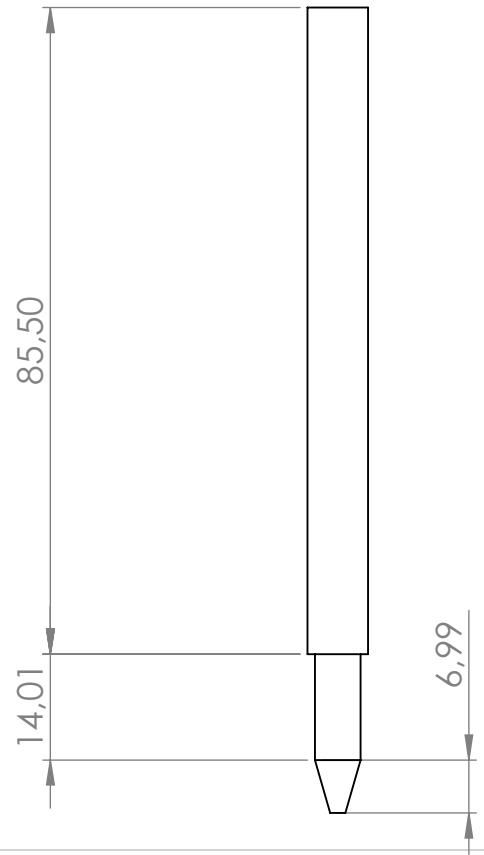
VISTA HORIZONTAL



VISTA FRONTAL



VISTA PERFIL



	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-11

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

PULSADOR

ESCALA

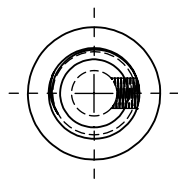
1:1

PLANO

N° 11

TORNILLO M12

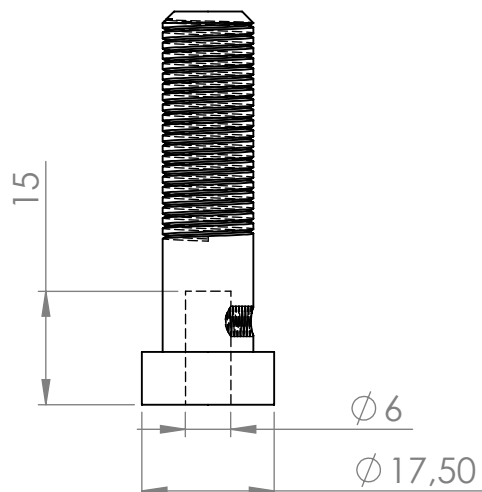
VISTA HORIZONTAL



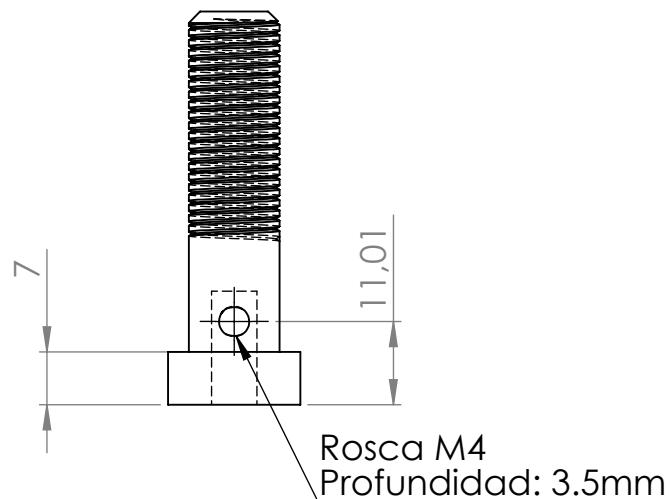
VISTA ISOMÉTRICA



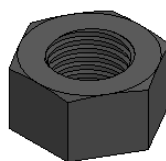
VISTA FRONTAL



VISTA PERFIL



TUERCA M12



	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-12

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

TORNILLO Y TUERCA
M12

ESCALA

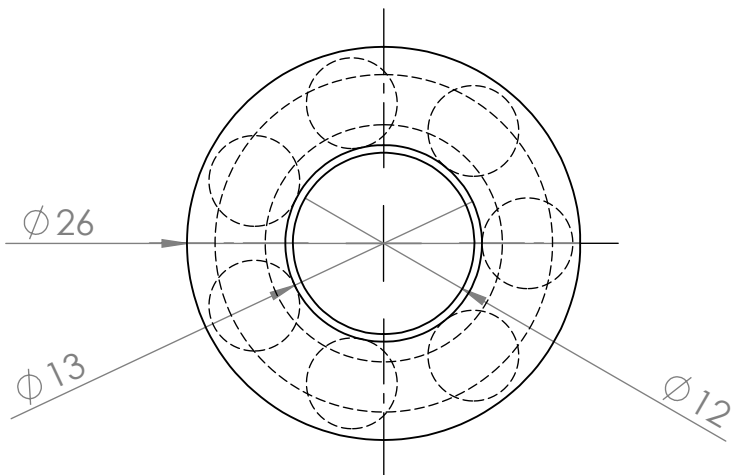
1:1

PLANO

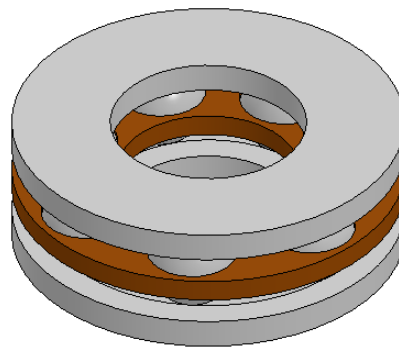
Nº 12

RODAMIENTO AXIAL 51101

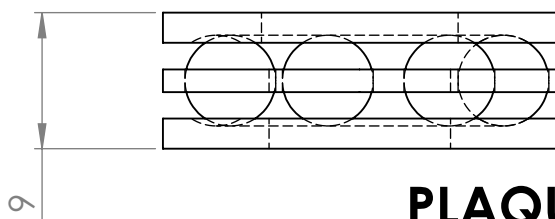
VISTA HORIZONTAL



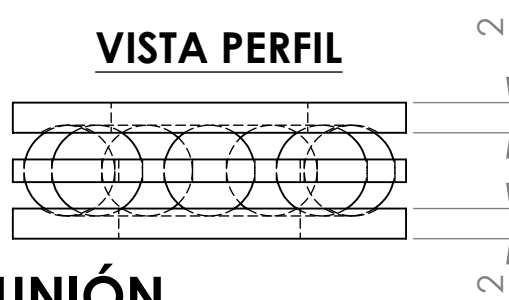
VISTA ISOMÉTRICA



VISTA FRONTAL

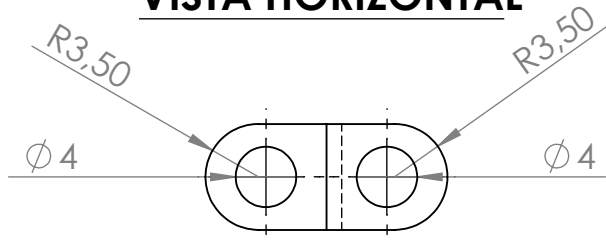


VISTA PERFIL

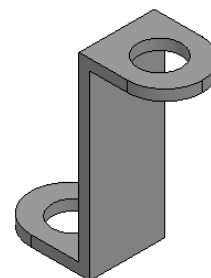


PLAQUITA DE UNIÓN

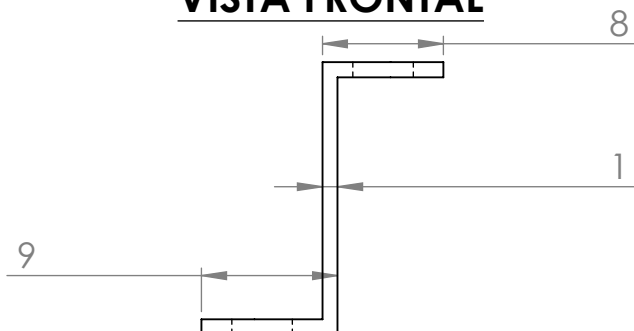
VISTA HORIZONTAL



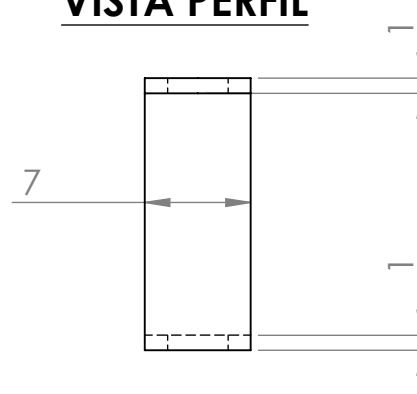
VISTA ISOMÉTRICA



VISTA FRONTAL



VISTA PERFIL



	FECHA	NOMBRE
DIBUJADO	01/10/16	MATR
REVISADO	01/10/16	MATR

UNIVERSIDAD DE PIURA
FACULTAD DE INGENIERÍA

TÍTULO

ANEXO A-13

NOMBRE Y APELLIDO

MARCO AURELIO TRONCOS RIOFRIO

RODAMIENTO AXIAL
Y
PLAQUITA DE UNIÓN

ESCALA

2:1

PLANO

Nº 13

**ANEXO B : CÁLCULOS DE LOS CENTROS DE
GRAVEDAD DE LAS PIEZAS DEL
BRAZO ROBOT RESPECTO A LA
MISMA PIEZA**

ANEXO B-1 : CENTRO DE GRAVEDAD DEL CUERPO

	Signo	Volúmen (mm ³)		x' (mm)	y' (mm)	z' (mm)	x'*V	y'*V	z'*V
1	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	-16,24	-16,24	-6,40	-16330,26	-16326,23	-6433,98
2	1	$24*10*12.8$	3072,00	0,00	-17,00	-6,40	0,00	-52224,00	-19660,80
3	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	16,24	-16,24	-6,40	16326,23	-16326,23	-6433,98
4	1	$44*105.5*12.8$	59417,60	0,00	52,75	-6,40	0,00	3134278,40	-380272,64
5	1	$44*58*12.8$	32665,60	0,00	99,10	29,00	0,00	3237160,96	947302,40
6	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	-16,24	99,10	62,24	-16330,26	99626,22	62574,52
7	1	$24*10*12.8$	3072,00	0,00	99,10	63,00	0,00	304435,20	193536,00
8	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	16,24	99,10	62,24	16326,23	99626,22	62574,52
9	-1	$(\pi)*(13^2)*(3)$	1592,79	0,00	0,00	-11,30	0,00	0,00	17998,53
10	-1	$(\pi)*(6.5^2)*(9.8)$	1300,78	0,00	0,00	-4,90	0,00	0,00	6373,82
11	-1	$(\pi)*(13^2)*(3)$	1592,79	0,00	104,00	46,00	0,00	-165650,16	-73268,34
12	-1	$(\pi)*(9^2)*(6.8)$	1730,39	0,00	99,10	46,00	0,00	-171481,65	-79597,94
13	-1	$(\pi)*(13^2)*(3)$	1592,79	0,00	94,20	46,00	0,00	-150040,82	-73268,34
14	-1	$(\pi)*(2^2)*(16)$	201,06	0,00	-14,00	-6,40	0,00	2814,85	1286,79
15	-1	$(\pi)*(2^2)*(6)$	75,40	11,30	96,70	28,06	-852,02	-7291,18	-2115,72
			110334,44				-860,06	6298601,58	650594,82

Coordenadas del centroide	X	-0,01
	Y	57,09
	Z	5,90

ANEXO B-2 : CENTRO DE GRAVEDAD DEL BRAZO

Item	Signo	Volumen (mm ³)		x' (mm)	y' (mm)	z' (mm)	x'*V	y'*V	z'*V
1	1	$(1/4) * \pi * (10^2) * (12.8)$	1005,31	-16,24	6,40	16,24	-16330,26	6433,98	16326,23
2	1	$24 * 10 * 12.8$	3072,00	-17,00	6,40	0,00	-52224,00	19660,80	0,00
3	1	$(1/4) * \pi * (10^2) * (12.8)$	1005,31	-16,24	6,40	-16,40	-16326,23	6433,98	-16487,08
4	1	$(1/4) * \pi * (10^2) * (12.8)$	1005,31	274,24	6,40	16,24	275700,24	6433,98	16326,23
5	1	$24 * 10 * 12.8$	3072,00	275,00	6,40	0,00	844800,00	19660,80	0,00
6	1	$(1/4) * \pi * (10^2) * (12.8)$	1005,31	274,24	6,40	-16,40	275700,24	6433,98	-16487,08
7	1	$282 * 44 * 12.8$	158822,40	129,00	6,40	0,00	20488089,60	1016463,36	0,00
8	-1	$(\pi) * (13^2) * (3)$	1592,79	0,00	1,50	0,00	0,00	-2389,19	0,00
9	-1	$(\pi) * (6.5^2) * (9.8)$	1300,77	0,00	7,90	0,00	0,00	-10276,08	0,00
10	-1	$(\pi) * (13^2) * (3)$	1592,79	258,00	1,50	0,00	-410939,82	-2389,19	0,00
11	-1	$(\pi) * (9^2) * (6.8)$	1730,39	258,00	6,40	0,00	-446440,62	-11074,50	0,00
12	-1	$(\pi) * (13^2) * (3)$	1592,79	258,00	11,30	0,00	-410939,82	-17998,53	0,00
13	-1	$(\pi) * (2^2) * (16)$	201,06	0,00	6,40	-14,00	0,00	-1286,79	2814,85
14	-1	$(\pi) * (2^2) * (6)$	75,40	239,82	4,00	-10,47	-18082,43	-301,60	789,44
			177073,63				20513006,89	1035805,03	3282,59

Coordenadas del centroide	X	115,84
	Y	5,85
	Z	0,02

ANEXO B-3 : CENTRO DE GRAVEDAD DEL ANTEBRAZO

Item	Signo	Volumen (mm ³)		x' (mm)	y' (mm)	z' (mm)	x'*V	y'*V	z'*V
1	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	-16,24	6,40	16,24	-16330,26	6433,98	16326,23
2	1	$24*10*12.8$	3072,00	-17,00	6,40	0,00	-52224,00	19660,80	0,00
3	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	-16,24	6,40	-16,40	-16326,23	6433,98	-16487,08
4	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	113,24	6,40	16,24	113845,33	6433,98	16326,23
5	1	$24*10*12.8$	3072,00	114,00	6,40	0,00	350208,00	19660,80	0,00
6	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	113,24	6,40	-16,40	113845,33	6433,98	-16487,08
7	1	$97*44*12.8$	54630,40	48,50	6,40	0,00	2649574,40	349634,56	0,00
8	-1	$(\pi)*(13^2)*(3)$	1592,79	0,00	1,50	0,00	0,00	-2389,19	0,00
9	-1	$(\pi)*(6.5^2)*(9.8)$	1300,77	0,00	7,90	0,00	0,00	-10276,08	0,00
10	-1	$(\pi)*(13^2)*(3)$	1592,79	97,00	1,50	0,00	-154500,63	-2389,19	0,00
11	-1	$(\pi)*(9^2)*(6.8)$	1730,39	97,00	6,40	0,00	-167847,83	-11074,50	0,00
12	-1	$(\pi)*(13^2)*(3)$	1592,79	97,00	11,30	0,00	-154500,63	-17998,53	0,00
13	-1	$(\pi)*(2^2)*(16)$	201,06	0,00	6,40	-14,00	0,00	-1286,79	2814,85
14	-1	$(\pi)*(2^2)*(6)$	75,40	80,79	8,80	-13,54	-6091,57	-663,52	1020,92
			72881,63				2659651,91	368614,31	3514,07

Coordenadas del centroide	X	36,49
	Y	5,06
	Z	0,05

ANEXO B-4 : CENTRO DE GRAVEDAD DE LA MANO

Item	Signo	Volúmen (mm ³)		x' (mm)	y' (mm)	z' (mm)	x'*V	y'*V	z'*V
1	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	-16,24	6,40	16,24	-16330,26	6433,98	16326,23
2	1	$24*10*12.8$	3072,00	-17,00	6,40	0,00	-52224,00	19660,80	0,00
3	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	-16,24	6,40	-16,40	-16326,23	6433,98	-16487,08
4	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	118,24	-22,64	0,00	118871,88	-22764,24	0,00
5	1	$24*10*12.8$	3072,00	119,00	6,40	0,00	365568,00	19660,80	0,00
6	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	118,24	9,44	0,00	118871,88	9494,15	0,00
7	1	$53*44*12.8$	29849,60	14,50	6,40	0,00	432819,20	191037,44	0,00
8	1	$15.6*10*12.8$	1996,80	46,00	6,40	14,20	91852,80	12779,52	28354,56
9	1	$15.6*10*12.8$	1996,80	46,00	6,40	-14,20	91852,80	12779,52	-28354,56
10	1	$15.6*10*12.8$	1996,80	46,00	20,60	0,00	91852,80	41134,08	0,00
11	1	$15.6*10*12.8$	1996,80	46,00	-7,80	0,00	91852,80	-15575,04	0,00
12	1	$63*44*12.8$	35481,60	82,50	6,40	0,00	2927232,00	227082,24	0,00
13	-1	$(\pi)*(13^2)*(3)$	1592,79	0,00	1,50	0,00	0,00	-2389,19	0,00
14	-1	$(\pi)*(6.5^2)*(9.8)$	1300,77	0,00	7,90	0,00	0,00	-10276,08	0,00
15	-1	$(\pi)*(13^2)*(3)$	1592,79	102,00	6,40	4,90	-162464,58	-10193,86	-7804,67
16	-1	$(\pi)*(9^2)*(6.8)$	1730,39	102,00	6,40	0,00	-176499,78	-11074,50	0,00
17	-1	$(\pi)*(13^2)*(3)$	1592,79	102,00	6,40	-4,90	-162464,58	-10193,86	7804,67
18	-1	$(\pi)*(2^2)*(16)$	201,06	0,00	6,40	14,00	0,00	-1286,79	-2814,85
19	-1	$(\pi)*(2^2)*(6)$	75,40	84,75	18,31	2,40	-6390,15	-1380,57	-180,96
			91569,63				3738074,57	461362,40	-3156,66

Coordenadas del centroide	X	40,82
	Y	5,04
	Z	-0,03

ANEXO B-5 : CENTRO DE GRAVEDAD DEL COMPONENTE FINAL

Item	Signo	Volúmen (mm ³)		x' (mm)	y' (mm)	z' (mm)	x'*V	y'*V	z'*V
1	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	-16,24	6,40	16,24	-16330,26	6433,98	16326,23
2	1	$24*10*12.8$	3072,00	-17,00	6,40	0,00	-52224,00	19660,80	0,00
3	1	$(1/4)*\pi*(10^2)*(12.8)$	1005,31	-16,24	6,40	-16,40	-16326,23	6433,98	-16487,08
4	1	$60.5*44*12.8$	34073,60	18,25	6,40	0,00	621843,20	218071,04	0,00
5	1	$(\pi)*(14^2)*(118.8)$	73151,36	31,50	59,40	0,00	2304267,84	4345190,78	0,00
6	-1	$(\pi)*(14^2)*(12.8)$	7881,63	31,50	6,40	0,00	-248271,35	-50442,43	0,00
7	-1	$(\pi)*(10^2)*(118.8)$	37322,12	31,50	59,40	0,00	-1175646,78	-2216933,93	0,00
8	-1	$(\pi)*(13^2)*(3)$	1592,79	0,00	1,50	0,00	0,00	-2389,19	0,00
9	-1	$(\pi)*(6.5^2)*(9.8)$	1300,77	0,00	7,90	0,00	0,00	-10276,08	0,00
10	-1	$(\pi)*(2^2)*(16)$	201,06	0,00	6,40	14,00	0,00	-1286,79	-2814,85
			160605,95				1417312,42	2314462,17	-2975,70

Coordenadas del centroide	X	8,82
	Y	14,41
	Z	-0,02

ANEXO B-6 : CENTRO DE GRAVEDAD DEL PULSADOR

Item	Signo	Volúmen (mm ³)		x' (mm)	y' (mm)	z' (mm)	x'*V	y'*V	z'*V
1	1	$\pi \cdot (4^2) \cdot (85.5)$	4297,70	0,00	0,00	42,75	0,00	0,00	183726,68
2	1	$\pi \cdot (3^2) \cdot (14.01)$	396,12	0,00	0,00	92,51	0,00	0,00	36643,08
3	1	$(1/\pi) \cdot (3^2) \cdot (6.99)$	65,90	0,00	0,00	101,26	0,00	0,00	6673,03
			4759,72				0,00	0,00	227042,79

Coordenadas del centroide	X	0,00
	Y	0,00
	Z	47,70

ANEXO C : DATOS TÉCNICOS DEL ENCODER INCREMENTAL

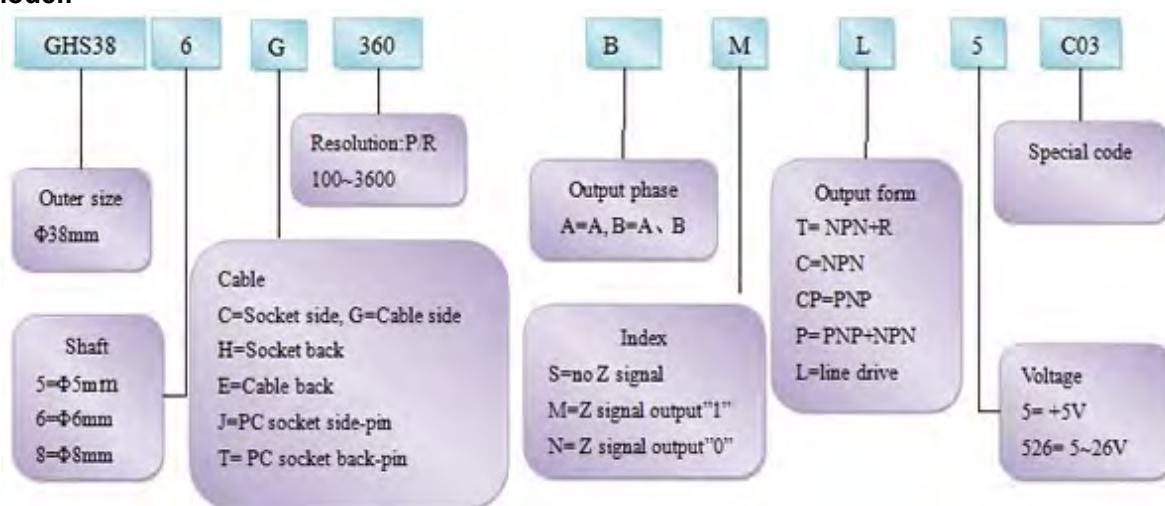
Solid shaft incremental rotary encoder- GHS38 series

Application & Features:

Widely used in automatic control, automatic measurement, as the angle and speed sensors. ASIC devices internal, good reliability, long life, anti-jamming performance. Stainless steel shaft has a higher stability and protection. Metal shell is more solid and anti-impact. Bearing load heavy, anti-oil and water.

1. ASIC devices internal, good reliability, long life, anti-jamming performance.
2. Stainless steel shaft has a higher stability and protection.
3. Metal shell is more solid and anti-impact.
4. Outlet end use waterproof metal interface protection
5. Bearing load heavy, anti-oil and water.
6. Applicability to a wide variety of mechanical and industrial control.
7. 100 ~ 3600 pulses Optional: (Other pulses available on request)
8. High grade protection is optional

Model:



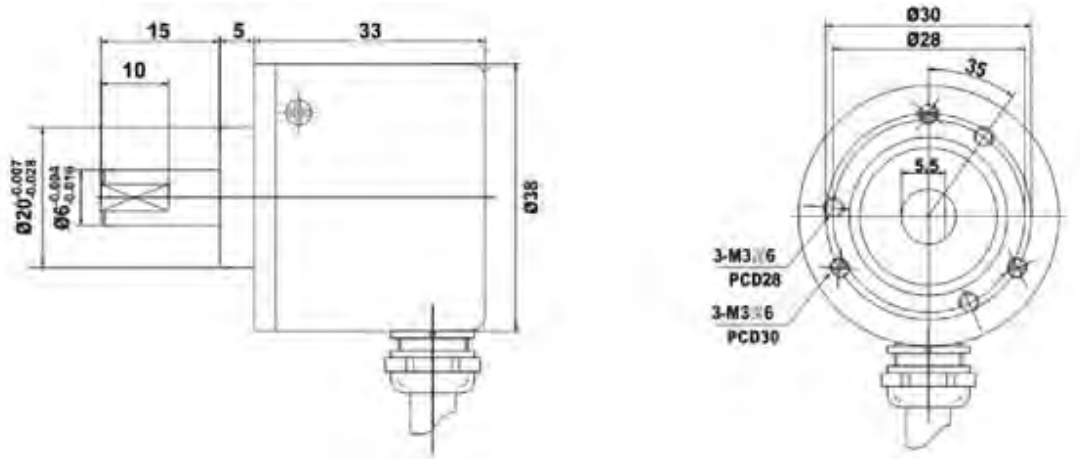
Electrical spec :

Output circuit	Open-collector output	Voltage output	Push-pull complementary outputs	driver
Supply voltage	5~30V	5~30V	5±0.25 5~30V	5±0.25 5~30V
Current consumption	≤80mA	≤80mA	≤80mA	≤150mA
Load current	40mA	40mA	40mA	60mA
Output high	Min Vcc*70%	Min Vcc-2.5V	Min Vcc-1.5V	Min 3.4V
Output low	MAX 0.4V	MAX 0.4V	MAX 0.8V	MAX 0.4V
Rise time Tr	MAX 1us	MAX 1us	MAX 1us	MAX 200ns
Fall time Tf	MAX 1us	MAX 1us	MAX 1us	MAX 200ns
Highest frequency response	300kHz	300kHz	300kHz	300kHz

Mechanical spec:

Max speed r/min	Starting torque	Axis maximum load	Impact	Anti-vibration
3000	<0.05Nm	Radial 25 N Axial 15N	50G/11ms	10G 10-2000Hz
Moment inertia 4 x 10 ⁻⁵ kgm ²	Work Temperature -30~85°C	Storage Temperature -35~85°C	Protection Degree IP54/IP65	Weight 100g

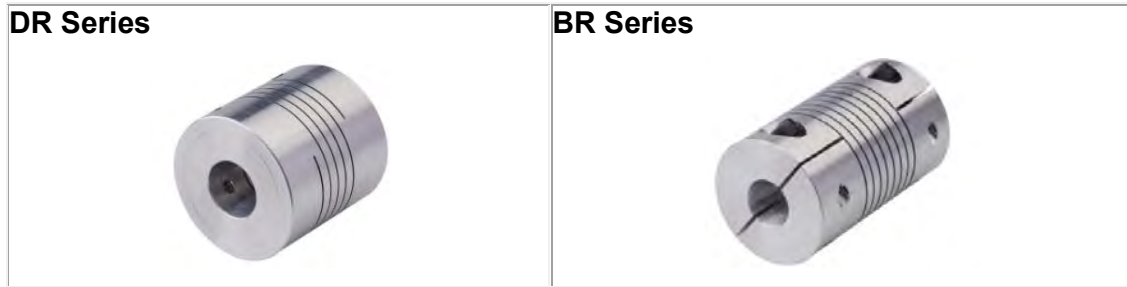
Dimensions :



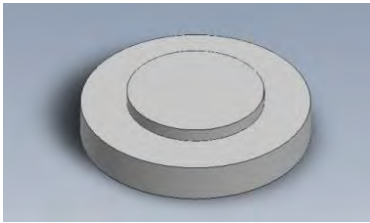
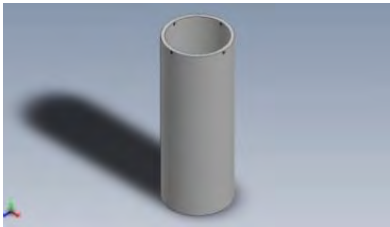
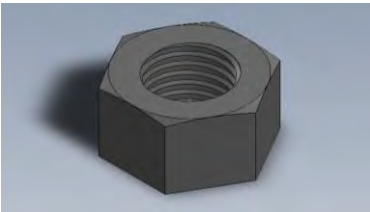
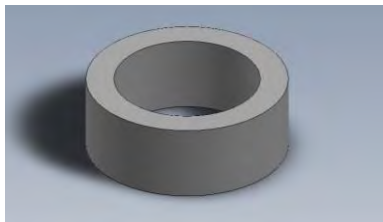
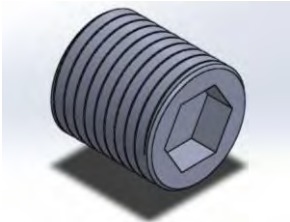
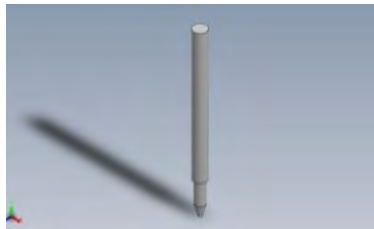
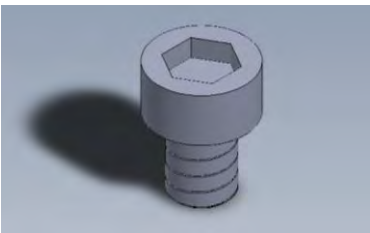
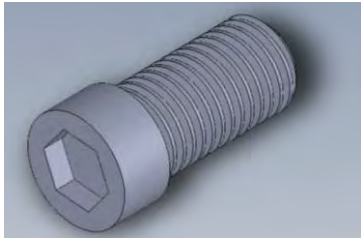
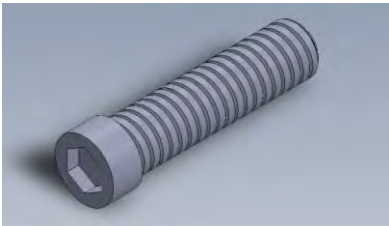
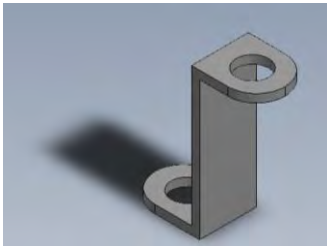
Connections:

Signal	A	B	Z	A-	B-	Z-	Vcc	GND
Color	Green	White	Yellow	Brown	Grey	Orange	Red	Black
Without inverted signal	Green	White	Yellow	-	-	-	Red	Black

Recommended of flexible coupling:



ANEXO D : PIEZAS SECUNDARIAS

DISCO BASE	CILINDRO BASE
	
TUERCA HEXAGONAL M12	BOCINA
	
PRISIONERO – TORNILLO M4 SIN CABEZA CON HUECO HEXAGONAL DE LONGITUD ADAPTADA A 3.5 mm	PULSADOR
	
TORNILLO M3 DE LONGITUD ADAPTADA A 4mm	TORNILLO M4 DE LONGITUD ADAPTADA A 6 mm
	
TORNILLO M4 DE LONGITUD ADAPTADA A 16 mm	PLAQUITA DE UNIÓN
	

ANEXO E : LISTADO Y COSTO DE MATERIALES

1	Placa Electrónica con Arduino Nano w/ATmega328	S/. 60,00	1	S/. 60,00
2	Encoder Incremental Serie GHS38	S/. 336,30	5	S/. 1.681,50
3	Rodamiento Axial 51101	S/. 5,00	10	S/. 50,00
4	Piezas de Ertalyte	S/. 939,57	1	S/. 939,57
5	Disco Base	S/. 10,00	1	S/. 10,00
6	Base	S/. 10,00	1	S/. 10,00
7	Pulsador	S/. 5,00	1	S/. 5,00
8	Plaquita de unión	S/. 0,50	5	S/. 2,50
9	Bocina	S/. 1,00	5	S/. 5,00
10	Trabajos de Mecanizado	S/. 60,00	1	S/. 60,00
11	Perno cabeza hexagonal M12 – Media rosca	S/. 1,50	5	S/. 7,50
12	Tuerca hexagonal M12	S/. 0,50	5	S/. 2,50
13	Tornillo cabeza hexagonal M4	S/. 0,30	14	S/. 4,20
14	Tornillo sin cabeza con hueco hexagonal M4 - Prisionero	S/. 0,80	5	S/. 4,00
15	Tornillo cabeza hexagonal M3	S/. 0,20	5	S/. 1,00
				S/. 2.842,77
Cotización de herramientas secundarias para el montaje				
1	Llave Allen para prisionero M4	S/. 4,00	1	S/. 4,00
2	Juego de Machos de roscar	S/. 15,00	2	S/. 30,00
3	Llave 17	S/. 20,00	1	S/. 20,00
4	Llave 19	S/. 20,00	1	S/. 20,00
5	Vernier de medición	S/. 35,00	1	S/. 35,00
				S/. 109,00
COTIZACIÓN TOTAL DEL BRAZO ROBOT				2.951,77