

Comparación entre un controlador PID y un controlador
CTC aplicado a un robot Puma de seis grados de libertad en
aplicaciones de *pick and place*



Universidad
del Cauca®

Edwin Geovanny Mena Barahona
Karen Viviana Perafán Camacho

Director:
PhD. Oscar Andrés Vivas Albán

Universidad del Cauca
Facultad de Electrónica y Telecomunicaciones
Programa de Ingeniería en Automática Industrial
Departamento de Electrónica, Instrumentación y Control
Popayán, Septiembre 2022

**Comparación entre un controlador PID y un controlador
CTC aplicado a un robot Puma de seis grados de libertad en
aplicaciones de *pick and place***

**Edwin Geovanny Mena Barahona
Karen Viviana Perafán Camacho**

Monografía presentada como requisito
para obtener el título de

Ingeniero en Automática Industrial

Director:

PhD. Oscar Andrés Vivas Albán

Universidad del Cauca

Facultad de Electrónica y Telecomunicaciones

Programa de Ingeniería en Automática Industrial

Departamento de Electrónica, Instrumentación y Control

Popayán, Septiembre 2022

Agradecimientos

Familia, amigos y personas especiales en nuestras vidas, no son nada mas y nada menos que un solo conjunto de seres queridos que suponen benefactores de importancia inimaginable en nuestras circunstancias de seres humanos, no podríamos sentirnos mas ajenos con la confianza puesta sobre nosotros, especialmente cuando hemos contado con su apoyo desde que tenemos memoria.

Este nuevo logro es en parte gracias a ustedes; hemos logrado concluir con éxito un proyecto que en un principio podría parecer tarea titánica e interminable, quisiéramos dedicar este trabajo de grado a ustedes, personas de bien, seres que ofrecen amor, bienestar y los finos deleites de la vida.

Muchas gracias a aquellos seres queridos que siempre guardamos en nuestra alma.

Resumen

En este proyecto se muestra el desarrollo, implementación y comparación de dos estrategias de control, PID (Proporcional, Derivativo e Integral) y CTC (Control por Torque Calculado) en un robot de seis grados de libertad, este trabajo de grado tiene como objetivo analizar e interpretar el funcionamiento de cada uno de estos métodos de control con el fin de verificar cuál de ellos brinda más eficacia a la hora de controlar este tipo de robots.

Para ello se aborda la construcción de un prototipo de robot Puma, a través del cual se busca hacer la comparación de cada una de las estrategias de control mencionadas y su posterior análisis cualitativo frente a varias trayectorias especialmente a una de tipo *Pick and Place*.

Como resultado se realiza una evaluación del desempeño de estos controladores, se observa y documenta mayor facilidad en la implementación de la estrategia CTC en contraposición con el control PID, ya que se logran desempeños superiores, bajo el supuesto, claro está, de conocer con relativa exactitud los parámetros dinámicos que caracterizan al robot.

Abstract

This project shows the development, implementation and comparison of two control strategies, PID (Proportional, Derivative and Integral) and CTC (Calculated Torque Control) in a robot with six degrees of freedom, this degree work aims to analyze and interpret the operation of each of these control methods in order to verify which of them provides more efficiency when controlling this type of robots.

For this, the construction of a Puma robot prototype is addressed, through which it is sought to make the comparison of each of the control strategies mentioned and its subsequent qualitative analysis against several trajectories, especially one of the Pick and Place type.

As a result, an evaluation of the performance of these controllers is carried out, it is observed and documented that it is easier to implement the CTC strategy as opposed to the PID control, since superior performances are achieved, under the assumption, of course, of knowing with relative accuracy the dynamic parameters that characterize the robot.

Contenido

Agradecimientos	2
Resumen	3
Abstract	4
Contenido	9
Índice de figuras	9
1 Introducción	9
2 Estado del arte	11
3 Diseño e implementación del prototipo	14
3.1 Diseño de piezas mediante software 3D CAD	14
3.1.1 Eslabón 1 (Cuerpo)	15
3.1.2 Eslabón 2 (hombro)	15
3.1.3 Eslabón 3 (brazo)	16
3.1.4 Eslabón 4 (antebrazo)	16
3.1.5 Eslabón 5 (Muñeca)	17
3.1.6 Eslabón 6 (Pinza)	17
3.2 Parámetros inerciales del robot Puma.	18
3.3 Impresión de eslabones	22
3.4 Elementos de medición y accionamiento	27
3.4.1 Sensores	27
3.4.2 Actuadores	28
3.5 Elementos de control	31
3.5.1 Tarjeta electrónica de control	31
3.6 Elementos de sistema de potencia	32
3.6.1 Fuente de alimentación	32
3.6.2 Módulos de control de motores	33
3.7 Conexión eléctrica de componentes	34
3.8 Análisis de la posición angular	36
3.9 Protocolos de comunicación	36
3.9.1 Protocolo de comunicación UART	36
3.9.2 Configuración del software de control	39
3.9.3 Consideraciones finales sobre el prototipo	41
3.9.4 Consideraciones del sistema de control	41

3.9.5	Consideraciones sobre los sensores	42
3.9.6	Consideraciones sobre el modelo 3D	42
4	Modelo del robot	43
4.1	Análisis matemático y diseño de controladores	43
4.1.1	Descripción de la estructura y modelo matemático	43
4.1.2	Modelos Geométricos	44
4.1.3	Modelo geométrico directo	45
4.1.4	Modelo geométrico inverso	48
4.2	Diseño de controladores	49
4.2.1	Control por par calculado	50
4.2.2	Esquema general del controlador CTC	50
4.2.3	Simulación en MATLAB - Simulink	50
4.2.4	Control PID	55
4.2.5	Esquema general del controlador PID	55
4.2.6	Simulación en MATLAB - Simulink	56
5	Pruebas y resultados	61
5.1	Resultados de implementación	61
5.1.1	Marco de referencia	62
5.1.2	Trayectorias establecidas	62
5.1.3	Trayectoria grado cinco	62
5.1.4	Trayectoria Circular	62
5.1.5	Trayectoria pick and place	63
5.2	Desempeño de control CTC y control PID	64
5.3	Desempeño de control CTC	65
5.3.1	Comportamiento robot real trayectoria grado cinco CTC	65
5.3.2	Comportamiento robot real trayectoria circular CTC	67
5.4	Desempeño control PID	68
5.4.1	Comportamiento robot real trayectoria grado cinco PID	68
5.4.2	Comportamiento robot real trayectoria circular PID	68
5.5	Desempeño de control CTC en trayectoria pick and place	69
5.5.1	Trayectoria pick and place con control CTC Matlab-Simulink	69
5.6	Desempeño de control PID en trayectoria pick and place	73
5.7	Análisis de velocidad	74
6	Resultados	75
6.1	Conclusiones	75
6.2	Trabajos futuros	75

Índice de figuras

3.1	Eslabón 1. Vista isométrica	15
3.2	Eslabón 1. Vista superior	15
3.3	Eslabón 2. Vista isométrica	16
3.4	Eslabón 2. Vista superior	16
3.5	Eslabón 3. Vista isométrica	16
3.6	Eslabón 3. Vista superior	16
3.7	Eslabón 4. Vista superior	17
3.8	Eslabón 4. Vista trimétrica	17
3.9	Eslabón 5. Vista dimétrica	17
3.10	Eslabón 5. Vista posterior	17
3.11	Pinza	18
3.12	Vista general del diseño en Solidwork 3D	18
3.13	Articulación 1 y sus propiedades físicas.	19
3.14	Articulación 2 y sus propiedades físicas.	20
3.15	Articulación 3 y sus propiedades físicas.	20
3.16	Articulación 4 y sus propiedades físicas.	21
3.17	Articulación 5 y sus propiedades físicas.	21
3.18	Articulación 6 y sus propiedades físicas.	21
3.19	Parámetros en Ultimake Cura.	23
3.20	Eslabón 1. Escenario Ultimake Cura.	23
3.21	Cálculo de tiempo de impresión.	24
3.22	Impresora Ender 3D.	24
3.23	Eslabón impreso 1. Vista frontal	25
3.24	Eslabón impreso 1. Vista superior	25
3.25	Eslabón 2. Escenario Ultimake Cura.	25
3.26	Eslabón impreso 2. Vista superior	25
3.27	Eslabón 3. Escenario Ultimake Cura.	25
3.28	Eslabón impreso 3. Vista superior	25
3.29	Eslabón 4. Escenario Dremel.	26
3.30	Impresora 3D Dremel.	26
3.31	Eslabón impreso 4. Vista superior	27
3.32	Eslabón 5 escenario Dremel.	27
3.33	Eslabón impreso 5. Vista isométrica	27
3.34	Potenciómetro Rotatorio Ky 040 incremental 360 Grados.	28
3.35	Motorreductor de 30:1 con Encoder POL1443.	29
3.36	Motorreductor Pelv.	30
3.37	Motorreductor DC MR12120.	30
3.38	Servomotor CDS5516.	31

3.39	Arduino MEGA 2560.	32
3.40	Fuente de voltaje S-180-12.	33
3.41	Puente H L298n	34
3.42	Puente H L298n	34
3.43	Conexión de Potencia	35
3.44	Conexión eléctrica	35
3.45	Diagrama de bloques de posición angular	36
3.46	Esquema protocolo UART	37
3.47	Formato de trama en UART	37
3.48	Comunicación Arduino a Matlab/Simulink.	38
3.49	Código de configuración Arduino a Matlab/Simulink	38
3.50	Bloques de lectura y escritura de datos.	39
3.51	Configuración bloque Digital read.	40
3.52	Configuración bloque Digital Write	41
4.1	Arquitectura robot Puma.	43
4.2	Sistema de coordenadas para articulación.	44
4.3	Esquema general control CTC.	50
4.4	Esquema control CTC cartesiano.	51
4.5	Control CTC para robot Puma.	51
4.6	Error cartesiano del control CTC para un robot Puma.	52
4.7	Trayectoria deseada vs trayectoria obtenida.	53
4.8	Esquema de control CTC articular.	53
4.9	Control CTC para robot Puma.	54
4.10	Error articular del control CTC para un robot Puma.	54
4.11	Esquema general control PID.	56
4.12	Esquema control PID cartesiano.	56
4.13	Control PID para robot Puma.	57
4.14	Error cartesiano PID.	58
4.15	Comportamiento de control PID cartesiano frente a una trayectoria circular.	59
4.16	Esquema PID articular robot Puma.	59
4.17	Control PID articular robot Puma.	60
4.18	Error articular PID articular.	60
5.1	Prototipo robot Puma terminado.	61
5.2	Trayectoria grado cinco.	62
5.3	Trayectoria Circular.	63
5.4	Trayectoria pick and place.	63
5.5	Error articular CTC con trayectoria grado cinco.	65
5.6	Trayectoria lineal.	66
5.7	Trayectoria lineal.	66
5.8	Trayectoria lineal.	66
5.9	Trayectoria lineal.	66
5.10	Error CTC Cartesiano del robot real con trayectoria circular.	67
5.11	Error PID articular del robot real con trayectoria grado cinco.	68
5.12	Error cartesiano PIS del robot real con trayectoria circular.	69
5.13	Trayectoria pick and place.	70

5.14	Trayectoria pick and place.	71
5.15	Trayectoria pick and place.	72
5.16	Error CTC cartesiano pick and place.	73
5.17	Trayectoria pick and place PID en Matlab-Simulink.	74

Capítulo 1

Introducción

Se han implementado diferentes estrategias como solución a los problemas relacionados con mejorar el rendimiento de controladores en el seguimiento de trayectorias, proponiendo una gran variedad de algoritmos y métodos convencionales de control en aplicaciones de robótica y áreas relacionadas, entre los cuales se destacan el control Proporcional Integral Derivativo (PID) y el Computer Torque Control (CTC). El presente trabajo de grado estudia comparativamente el desempeño de estas dos estrategias de control.

Las estrategias de control son algoritmos encargados de determinar las acciones necesarias para llevar un sistema a un estado deseado, partiendo de la información disponible sobre el estado actual y/o las características del mismo. La estrategia de control PID, es ampliamente usada en el sector industrial por su efectividad y simplicidad en el control de sistemas [1], teniendo en cuenta que la sintonización de este tipo de controladores tiene una mayor complejidad cuando se hace manualmente, por esto los métodos mas usados son: redes neuronales, fuzzy, entre otros. Por otra parte los robots de estructura serial están relacionados por una cadena cinemática abierta, presentan una configuración de eslabones conectados en forma secuencial, empezando por la base hasta el efector final.

Cada eslabón de la cadena está unido al anterior mediante articulaciones, ya sean rotacionales o prismática, y en todas las articulaciones hay un generador de movimiento o actuador. Los motores de los actuadores están montados en las articulaciones de los diferentes eslabones que componen el robot, lo que hace que la mayor parte de los motores tengan ciertos desplazamientos móviles aumentando las masas y con ello las fuerzas de inercia y el riesgo de averías [2]. Robots con esta estructura resultan valiosos para operaciones donde la velocidad y la precisión sobre objetos pequeños son características críticas, como las de re-posicionamiento de objetos *pick and place*.

En la década de los años 30 del siglo XX (1937) el estudiante británico Bill Taylor creó el robot Gargantua, un robot con forma de grúa para el que utilizó componentes que se usaban en los juguetes de Meccano. En los años posteriores Unimation y Vicarm lanzaron al mercado el brazo robótico Puma, ideado por Scheinmann casi 10 años antes. En un comienzo fue desarrollado para las líneas de ensamblaje de General Motors [3].

En la actualidad, los robots industriales han tenido un gran avance a nivel industrial. Existen robots capaces de ejecutar diferentes tipos de acciones como son el seguimiento de trayectorias, tareas de apilamiento y ordenamiento de objetos, todo esto sobre la base de estudio de la cinemá-

tica directa e inversa determinada a través de la geometría del robot [4].

Entre los más importantes para realizar estas actividades se tiene a los robots de cinco grados de libertad ya que cada articulación puede ser de desplazamiento, de giro o una combinación de ambos [5], pero dado que las industrias y sus líneas de producción evolucionan y se encuentran en un constante cambio apuntando hacia la optimización, requieren más actividades y un robot de este tipo ya no sería suficiente. Según lo anterior por las desventajas descritas, los más utilizados en la línea de producción, entonces, son los robots de manipulación de seis grados de libertad o robots tipo Puma.

En la Universidad del Cauca el grupo de Automática Industrial cuenta con algunos robots tipo educativo como lo es Kit Robot R700 Vector Robotic Arm, Kit Robot SainSmart 5-Axis y Kit Robot SainSmart DIY 6-Axis DOF que han ayudado a la retroalimentación de los estudiantes en cuanto al manejo de las tareas de *pick and place*.

Debido a esto y por su importancia, en el presente proyecto se realiza el diseño e implementación de un controlador PID y un controlador CTC en un prototipo experimental tipo Puma y se efectúa un estudio del desempeño para definir cual es el más óptimo. Para ello el documento se encuentra dividido en seis capítulos.

El primer capítulo se presenta una breve explicación de lo que se desarrolla en el proyecto. El segundo capítulo presenta el estado del arte donde se da a conocer la evolución de los robots a medida que avanza el tiempo y su actualización.

En el capítulo tercero se incluye la implementación del sistema robótico deseado: el ensamblaje, los sistemas de control. Se presentan los eslabones que fueron diseñados en el software Solidworks obteniendo las propiedades físicas de cada eslabón como son: masa, centro de masa, inercia, etc; se analiza y establece las medidas de los motores que se usarán para su funcionamiento y el material para realizar la impresión, se especificaran los elementos de accionamiento y control a usar, se detallara los protocolos de comunicación a implementar.

En el cuarto capítulo se realiza el análisis cinemático y dinámico de la estructura robótica a implementar; describiendo aspectos como: la estructura, la geometría, la construcción de modelos matemáticos y el diseño de los controles a comparar, también se da a conocer su esquema general y la simulación en el software Matlab teniendo en cuenta los componentes de los eslabones obtenidos del diseño en Solidworks del robot Puma con sus debidos parámetros obtenidos del diseño en Solidworks.

En el capítulo cinco se detalla los resultados obtenidos tras la implementación del control PID y CTC sobre el prototipo implementado, se desarrolla el análisis comparativo del control para las trayectoria deseada. En el capítulo seis quedan consignadas las conclusiones del proyecto y los trabajos futuros.

Capítulo 2

Estado del arte

El avance exponencial de la tecnología ha permitido grandes avances en el campo de la robótica, esto se debe en gran parte al aumento de la capacidad de computo y a las características atractivas de los robots en el campo industrial, estos son muy utilizados en plantas de manufactura, montaje y embalaje, en transporte, en exploraciones, en la tierra y en el espacio, cirugía, armamento, investigación en laboratorios y en la producción en masa de bienes industriales o de consumo.

Como se menciona anteriormente, el desarrollo de la ciencia y de la tecnología ha promovido el avance de la robótica, convirtiéndola en un elemento de la transición actual [6]. En la era contemporánea, los sistemas y maquinaria, incluso la forma de producir elementos en la industria han cambiado, utilizando la robótica para brindar un mejor control en el proceso.

En ese proceso investigativo [7], muestra la importancia de los índices de desempeño para la planificación de movimientos y el diseño de robots manipuladores lo que ha dado a conocer una colección de ciertos puntos de interés para la comunidad dedicada a la robótica. Se presentan índices de desempeño cinetostático, dinámico, límites articulares e índices definidos sobre el espacio de trabajo.

Como se puede observar se tienen métodos de control para la construcción y optimización de robots manipuladores, pero el mundo moderno no se conforma con la punta del iceberg en cuanto a avances tecnológicos, es por ello que las aplicaciones robóticas han mostrado una tendencia hacia la precisión y movimiento de alta velocidad para lograr una tarea específica. Sin embargo, la eficiencia de los robots industriales se ve severamente reducida por la complejidad de su operación.

En términos matemáticos, la planificación y el control del movimiento de los robots es una tarea muy pesada dada la carga computacional para ser ejecutada en tiempo real. Los problemas en el control de los robots surgen de las vastas complejidades computacionales asociadas con la formulaciones, además de la necesidad de métodos de control adaptativos apropiados para lograr la precisión y la velocidad requeridas. Esta tendencia hacia los sistemas inteligentes es lo que lleva a experimentar con nuevas formas de control, entendiendo la necesidad de adaptar la robótica a las necesidades humanas avanzadas. Como ejemplo se tiene el diseño y desarrollo de un sistema de control para el movimiento teledirigido de un brazo robótico de 4 grados de libertad (DOF), aplicando visión de máquina [8]. Las pruebas del sistema teledirigido y del sistema de visión de máquina demostraron la exactitud del método para el cálculo de distancias espaciales, con un error bajo, respecto a las distancias medidas reales dentro del entorno de trabajo del manipulador.

Tradicionalmente se ha abordado el control de movimiento de los robots con estrategias como el PID que tiene una gran presencia en el sector industrial [9]. Sin embargo, esta estrategia desaprovecha características como las altas velocidades de los robots al introducir vibraciones en estado transitorio e inexactitudes en el estado estacionario [10]. En el esfuerzo por establecer un control con capacidad de abordar gran parte de los aspectos de funcionamiento de un sistema robótico se han desarrollado opciones que toman en cuenta el modelo dinámico del sistema como el CTC (control por torque computado) que se ha extendido por considerarse una estrategia confiable y precisa.

Pero el problema del seguimiento de trayectoria en robots manipuladores va más allá de solo poder adjuntar con éxito la trayectoria deseada con un control determinado. Una de las tantas tareas fundamentales que cumple el robot Puma es el seguimiento de trayectorias; el problema del seguimiento de trayectoria en robots manipuladores ha sido abordado aplicando una gran variedad de controladores, desde estructuras sencillas basadas en PD [11], pasando por los controladores CTC y llegando a otras más complejas basadas en controladores adaptativos y robustos. Estas últimas técnicas presentan inconvenientes como la presunción de ciertas cotas en los términos de la ecuación dinámica del robot.

El problema de controlar el movimiento de robots manipuladores ha sido abordado tradicionalmente con controladores PID, controladores PD con compensación de gravedad o técnicas basadas en control dinámico inverso [12]. Incluso han sido propuestas estrategias basadas en control difuso.

Torres y Méndez [13] muestran un trabajo hecho con un robot Puma 560 donde se hace una revisión de las técnicas clásicas de control de manipuladores y en el cual se introduce un conjunto de técnicas novedosas de control robusto y de control predictivo, con las que se evitan los problemas comentados. En particular se describe un controlador con una acción robusta auto adaptativa, necesaria para evitar los errores en la cancelación de términos no lineales de la dinámica del robot.

Los métodos de linealización se utilizan para superar los problemas de no linealidad que caracterizan a los robots manipuladores. Bali [14], presenta dos enfoques basados en el control predictivo generalizado. La primera consiste en aplicar esta estrategia de control al modelo no lineal de un robot manipulador y estudiar su robustez frente a variaciones en los parámetros de síntesis. El segundo enfoque se basa en la síntesis de varios reguladores locales y el cambio entre ellos.

Para desarrollar un control que considere el modelo dinámico del sistema se han desarrollado investigaciones donde podemos ver que los controladores PID y difuso no tienen en cuenta la dinámica del sistema, pero sólo son excepciones, porque se utilizan únicamente para resolver el problema de regulación, por ejemplo, el movimiento de un punto a otro punto sin imponer el seguimiento de una trayectoria específica. El resto de controladores, empleados para solucionar el problema del seguimiento de trayectoria, utilizan el modelo dinámico para calcular la acción de control. Así, por ejemplo, con el modelo se evalúa el término de gravedad en el caso de compensación de gravedad o el término no lineal en el caso de control dinámico inverso [15].

Por otro lado se tiene el controlador de torque computarizado (CTC) que es un poderoso controlador no lineal que se utiliza ampliamente en control de robots manipuladores. Se basa en la linealización de retroalimentación y calcula el torque requerido usando la ley de control de retroalimentación no lineal. Este controlador funciona muy bien cuando se conocen todos los parámetros dinámicos y físicos, pero no cuando el robot tiene variación en parámetros dinámicos,

en esta situación el controlador no tiene un rendimiento aceptable [16]. En la práctica, la mayoría de los parámetros de los sistemas físicos son desconocidos o variantes en el tiempo, por lo tanto, el controlador par de torque calculado es usado para compensar la ecuación dinámica del robot.

La investigación sobre el controlador de torque computarizado está creciendo significativamente en la aplicación de robots manipuladores [17][18].

Vivas y Mosquera han propuesto un controlador funcional predictivo el cual comparan con el controlador de torque computarizado para rastrear la respuesta en un ambiente incierto [19]. Sin embargo, en ambos controladores se ha utilizado la linealización y la retroalimentación, pero la estrategia predictiva da mejores resultados, tales como: rendimiento, eficacia y mejor desempeño.

Siguiendo esta línea de investigación se encuentra un estudio importante dentro de la robótica, el que realiza Yenorkar y Chaskar [20], en donde además de reconocer la importancia que tiene la tecnología y la robótica en la actualidad, los investigadores desarrollan un brazo robótico, entrenable y además basado en GUI que está siendo automatizado para aplicaciones industriales multipropósito. El aporte científico es que el robot se caracteriza por ser un brazo que se manipula fácilmente y tiene soluciones para cierta gama de aplicaciones de *pick and place*, la metodología científica se basa en el uso de la herramienta MATLAB y también en microcontroladores AVR. Este microcontrolador de la familia de microcontroladores RISC fue programado en la estación central de control que tiene acceso a GUI y también al control del brazo, además posee un programador para GUI para hacer que el robot responda a las órdenes siendo posible entrenar con el.

En el año 2019, se encuentran varias investigaciones que apoyan el actual trabajo de estudio, y que aportan teórica y metodológicamente a un desarrollo óptimo investigativo. Primero es importante mencionar la investigación [21], en el que se relata que los robots *pick and place* se utilizan con frecuencia en la industria para diferentes aplicaciones [21], se busca implementar un diseño basado en modelos para una aplicación de *pick and place* como un estudio de caso de la industria 4.0. Para ello, la investigación usó un brazo robótico universal UR5, que es una metodología que disminuye errores de comunicación y manejo innecesario de los proyectos cuando se ajuste un componente, como también de los costos y el tiempo, lo que permite ampliar la brecha de investigaciones, ya que muchas investigaciones se obstaculizan por falta de tiempo y recursos.

Finalmente, es importante resaltar la investigación [22], trabajo de gran importancia, en el cual que busca evaluar algoritmos de rastreo de objetivos de seis grados de libertad. Según los autores la justificación de la investigación aparece porque los datos existentes muestran grandes limitaciones, pues son poco realistas, que impiden a la comunidad obtener una imagen precisa del estado de la técnica. Para cumplir el objetivo los autores usaron una red de datos basada en un sistema comercial de captura de movimiento para adquirir posiciones exactas. se utiliza entonces una cámara Kinect V2 con la cual se adquiere tres escenarios diferentes para evaluar el rendimiento de los seguidores: estabilidad, robustez a la oclusión y precisión durante las complejas interacciones entre una persona y el objeto [22].

Capítulo 3

Diseño e implementación del prototipo

3.1. Diseño de piezas mediante software 3D CAD

En este capítulo se detalla de manera secuencial las necesidades que fueron surgiendo para diseñar el prototipo y a su vez, las medidas que se tomaron para resolverlas. De igual forma se mencionan los objetivos planteados en un principio para obtener los resultados deseados en esta etapa del trabajo. Se presentan las características de cada eslabón, el proceso del diseño y el proceso de impresión.

Dicho lo anterior, se analizará rápidamente el robot Puma, el nombre del brazo robótico Puma representa las siglas de “Programmable Universal Machine for Assembly, or Programmable Universal Manipulation Arm”. Este es un brazo robótico industrial creado por Victor Scheinman en la compañía robótica Unimation. Inicialmente desarrollado para General Motors [23].

El prototipo inicial del Puma 560, consistía en un robot de seis grados de libertad con muñeca esférica, con una capacidad de carga máxima de entre 2.5 Kg a 4 Kg, con un peso global aproximado de 83 Kg.

El Puma 560 ha sido estudiado y usado en innumerables experimentos e industrias. Sin embargo, sigue siendo un reto reunir completamente todos los datos necesarios para construir debidamente un modelo, ya que, al haber sido frecuentemente estudiado, cada uno de los estudios ha extraído sus propios datos y conclusiones. Es por ello por lo que, en el desarrollo de este proyecto, se ha decidido elegir la solución de Paul 81 debido a que es el estudio que se ha determinado más próximo a lo que podría ser la realidad.

Teniendo en cuenta lo anterior el diseño del robot Puma se realizó en el software de diseño CAD Solidworks. Este software permite la representación de piezas y conjuntos en 3D, además que admite trabajar sobre los siguientes tipos de entorno: pieza, conjunto, chapa, soldadura y plano. El diseño del prototipo de este proyecto se llevo a cabo en el entorno "pieza" para darle la forma deseada a cada uno de los componentes del robot con las medidas adecuadas. Posteriormente se ensamblan las piezas en el entorno "conjunto" obteniendo un diseño preliminar del Puma 560.

Para pasar a la impresión se consideró el material a utilizar ya que al hacer esto permite tener un idea de la cantidad necesaria de materia prima y también permite obtener las propiedades físicas de cada eslabón. Lo anterior servirá para la obtención de valores inerciales y por ende un resultado

óptimo a la hora de la simulación del modelo.

El material que se escogió en el software fue ABS (acrilonitrino, butadieno y estireno), el cual es el material mas parecido al material real de impresión el cual fue PLA (ácido poliláctico), también llamado ácido poliláctico, se deriva de materias primas naturales y renovables, como el maíz, y pertenece a los poliésteres como un polímero sintético. Como se menciona este proceso de asignación de material es muy importante para obtener unos valores inerciales correctos y que representen fielmente el modelo que queremos simular, por esta razón también se diseñan los motores con un peso y unas dimensiones aproximadas a las características brindadas.

Las dimensiones y medidas principales de dicho robot, son establecidas por criterios universales, así como el posicionamiento y orientación de los sistemas de referencia de cada una de sus barras y sus variables articulares (siguiendo el criterio de Denavit Hartenberg). El modelo guía para el diseño solo se presenta hasta el quinto eslabón, esto se debe a que el elemento final que es la pinza o griper en este caso, puede tener muchas variantes, estas pueden ser: piezas de sujeción o simplemente terminales para desarrollar diferentes tareas, esto depende del área de trabajo del robot.

Teniendo en cuenta lo descrito anteriormente se procedió al diseño del prototipo.

3.1.1. Eslabón 1 (Cuerpo)

Esta pieza es diseñada para garantizar la sujeción estable de todo el prototipo, capaz de ofrecer rigidez mecánica y estructural para un soporte fiable del hombro, en su interior se ubicara el motor para así formar como tal la primera articulación.

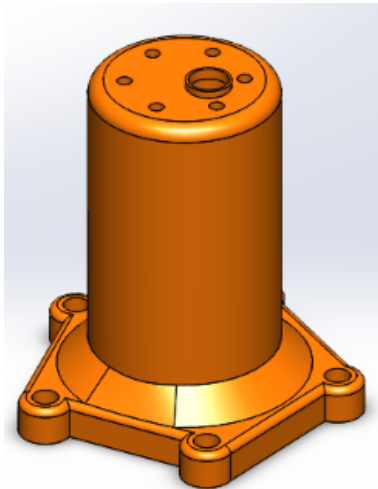


Figura 3.1: Eslabón 1. Vista isométrica
Fuente: Diseño propio.

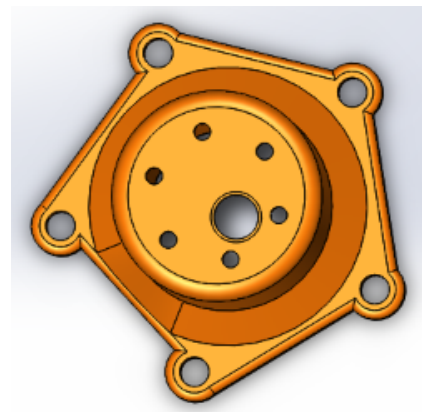


Figura 3.2: Eslabón 1. Vista superior
Fuente: Diseño propio.

3.1.2. Eslabón 2 (hombro)

Este elemento es rotatorio, está diseñado para ser conectado por medio de una articulación al siguiente eslabón, en nuestro caso esta articulación es el motor, que como en nuestra pieza anterior

también se va a ubicar en el interior de la pieza, de esta manera se va a lograr en conjunto un movimiento rotatorio integral.

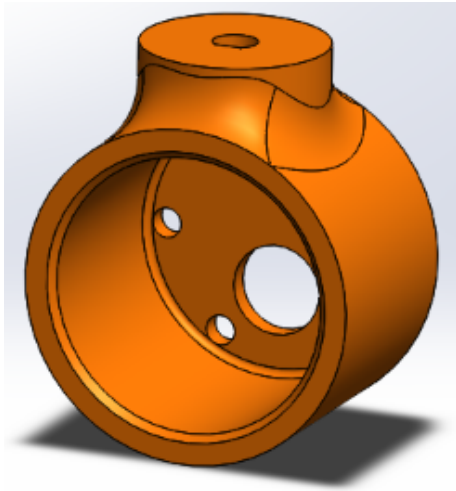


Figura 3.3: Eslabón 2. Vista isométrica
Fuente: Diseño propio.

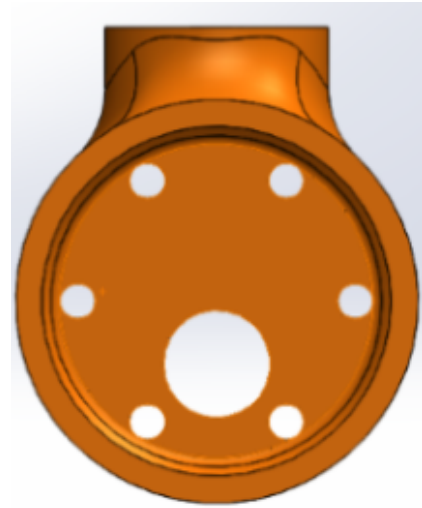


Figura 3.4: Eslabón 2. Vista superior
Fuente: Diseño propio.

3.1.3. Eslabón 3 (brazo)

Este elemento es rotatorio en sus extremos, teniendo en primera medida la parte que conecta con la segunda articulación y por otro lado se encuentra la parte de la pieza que conecta con el eslabón cuatro, las dos partes son perpendiculares a los respectivos ejes con los que conectan.

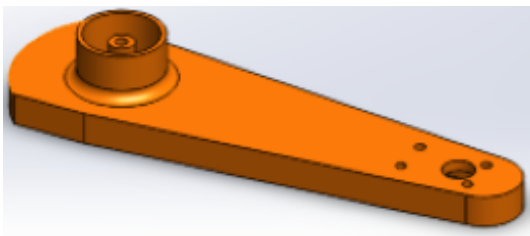


Figura 3.5: Eslabón 3. Vista isométrica
Fuente: Diseño propio.



Figura 3.6: Eslabón 3. Vista superior
Fuente: Diseño propio.

3.1.4. Eslabón 4 (antebrazo)

Como se menciona esta pieza hace la labor del antebrazo del robot, es un elemento de naturaleza rotatoria en la parte de conexión con el eslabón 3, así mismo en su parte final dispone de un vaciado para lograr incorporar el actuador (motor) del eslabón 5.

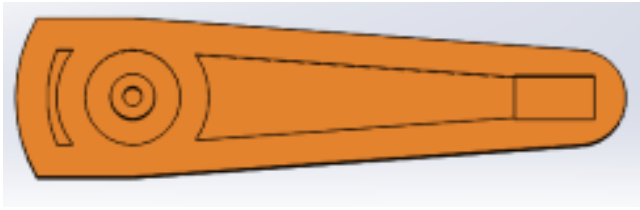


Figura 3.7: Eslabón 4. Vista superior
Fuente: Diseño propio.

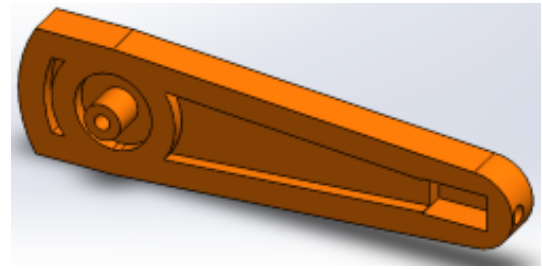


Figura 3.8: Eslabón 4. Vista trimétrica
Fuente: Diseño propio.

3.1.5. Eslabón 5 (Muñeca)

El eslabón mostrado en la siguiente figura es definido como la muñeca del prototipo, esta pieza también posee un movimiento que será de tipo rotatorio y así mismo este se conecta con la pinza del robot por medio de la articulación

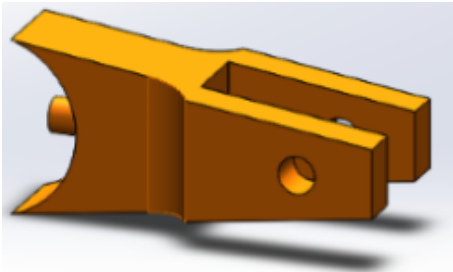


Figura 3.9: Eslabón 5. Vista dimétrica
Fuente: Diseño propio.

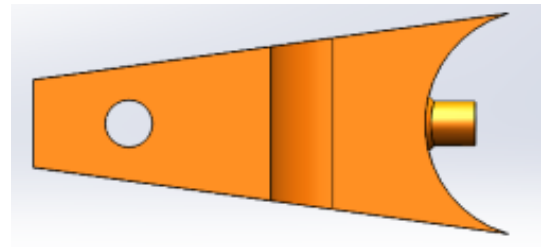


Figura 3.10: Eslabón 5. Vista posterior
Fuente: Diseño propio.

3.1.6. Eslabón 6 (Pinza)

Los elementos de sujeción más comunes son las denominadas «pinzas» o «garras». Este elemento se usó como accionamiento neumático para sujetar las piezas por presión y moverlas de lugar. Es el eslabón final del prototipo Puma.

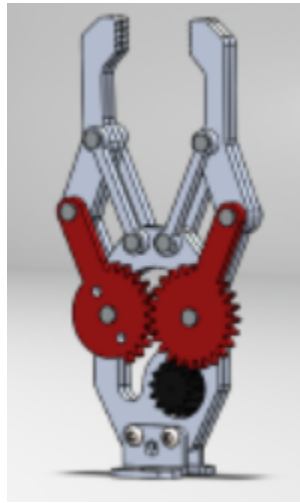


Figura 3.11: Pinza

Fuente: Diseño tomado de software Solidworks.

Una vez diseñado cada una de las piezas se realiza el ensamblaje con sus debidos motores, el prototipo final se muestra a continuación:

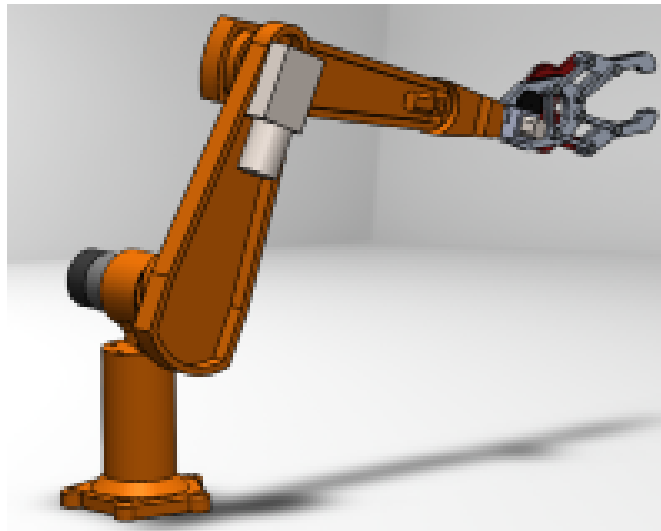


Figura 3.12: Vista general del diseño en Solidwork 3D

Fuente: Diseño propio.

3.2. Parámetros inerciales del robot Puma.

En primera instancia se debe generar la tabla de parámetros geométricos con el fin de: a partir de esta, encontrar el modelo geométrico directo (MGD), modelo dinámico directo (MDD) y modelo dinámico inverso (MDI). Existen once parámetros que definen la dinámica del robot por cada articulación las cuales son:

- Seis términos del tensor de inercia: XX_j , XY_j , XZ_j , YY_j , YZ_j , ZZ_j .
- Tres términos del primer momento de inercia: MX_j , MY_j , MZ_j .

- Un término para la masa: M_j .
- Un término para la inercia del motor: I_{aj} .

Estos parámetros serán calculados de manera simple por medio del software Solidworks. El objetivo sera obtener los valores numéricos más cercanos a los reales con el fin de poder realizar la simulación del sistema robot en un software de simulación como Matlab/Simulink. Sin estos valores numéricos no será posible ejecutar la simulación. de cualquiera de estos. El más importante de ellos es el centro de gravedad que se define como un punto fijo de un segmento donde actúa la fuerza resultante de la atracción de la gravedad.

Para conseguir los valores, se tiene en cuenta las propiedades físicas del modelo simulado en 3D, de esta manera logramos obtener los parámetros inerciales de cada articulación como se observa a continuación:

Teniendo en cuenta el centro de gravedad para cada una de las imágenes el software permite los parámetros inerciales asociados en secuencia de cada articulación:

$XX1=0.00080$, $XY1=-0.000043$, $XZ1= 0$, $YY1=0.000082$, $YZ1=-0$, $ZZ1=0.0008041$, $M1=0.25$, $MX1= 0.25*(-0.0036)$, $MY1=0.25*(0.048)$, $MZ1=0.25*(-0.00021)$, $IA1= 0.053$.

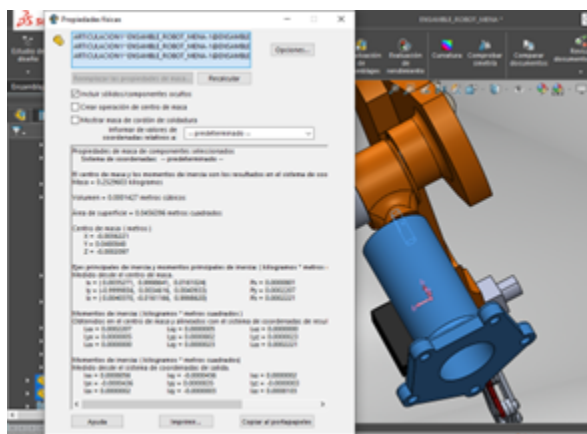


Figura 3.13: Articulación 1 y sus propiedades físicas.

Fuente: Diseño propio.

En esta siguiente imagen se visualiza los parámetros inerciales asociados a la segunda articulación: $XX2=0.003$, $XY2=-0.000273$, $XZ2=-0$, $YY2= 0.000111$, $YZ2= 0.0000025$, $ZZ2= 0.00346$, $MX2= 0.196*(-0.010)$, $MY2= 0.196*(0.130)$, $MZ2= 0.196*(0)$, $M2= 0.196$, $IA2= 0.053$.

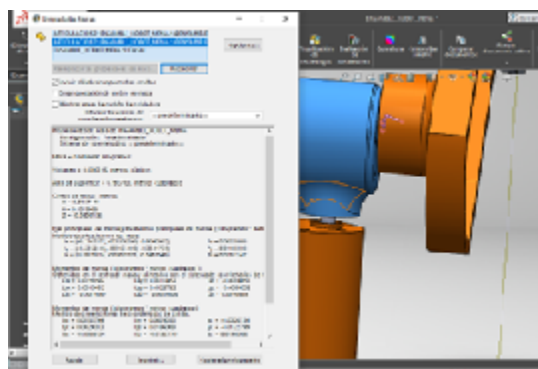


Figura 3.14: Articulación 2 y sus propiedades físicas.
Fuente: Diseño propio.

Esta tercera imagen permite visualizar los parámetros inerciales asociados a la tercera articulación: $XX3= 0.0075$, $XY3= 0.0018$, $XZ3= -0.00060$, $YY3= 0.001513$, $YZ3= -0.00226$, $ZZ3= 0.00709$, $MX3= 0.174*(0.0542)$, $MY3= 0.174*(0.189)$, $MZ3= 0.174*(-0.05)$, $M3= 0.174$, $IA3=0.018$.

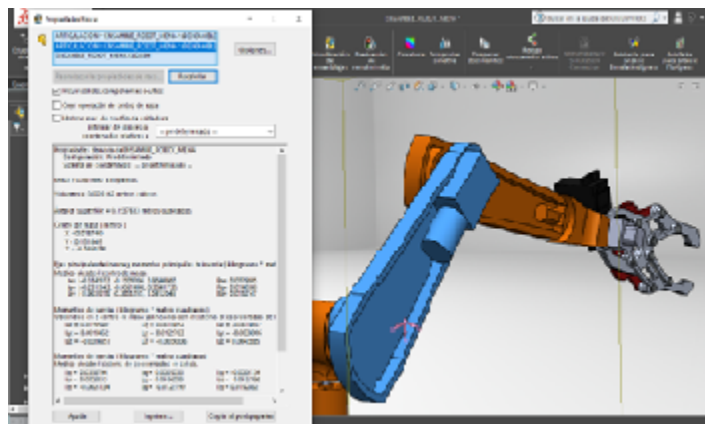


Figura 3.15: Articulación 3 y sus propiedades físicas.
Fuente: Diseño propio.

Esta cuarta imagen permite visualizar los parámetros inerciales asociados a la cuarta articulación: $XX4= 0.00617$; $XY4= 0.00057$; $XZ4= -0.00044$; $YY4= 0.0024$; $YZ4= -0.00268$; $ZZ4= 0.0037$; $MX4= 0.080*(0.0340)$; $MY4= 0.080*(0.2116)$; $MZ4= 0.080*(-0.1629)$; $M4= 0.080$; $IA4= 0.035$.

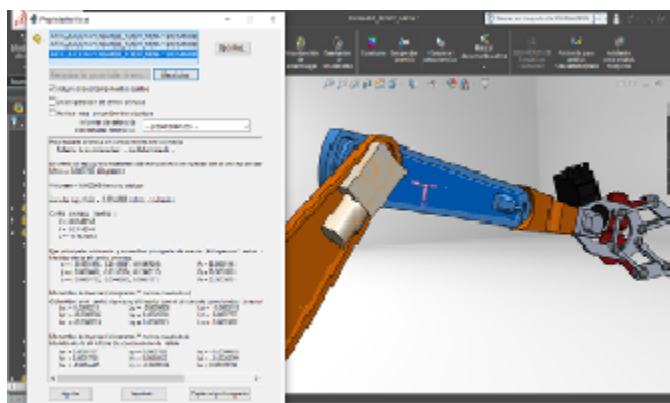


Figura 3.16: Articulación 4 y sus propiedades físicas.
Fuente: Diseño propio.

La siguiente imagen permite visualizar los parámetros inerciales asociados a la quinta articulación:
 $XX5 = 0.0038$; $XY5 = 0.00030$; $XZ5 = -0.00049$; $YY5 = 0.0028$; $YZ5 = -0.0017$; $ZZ5 = 0.00114$; $MX5 = 0.034 \cdot (0.034)$; $MY5 = 0.034 \cdot (0.2116)$; $MZ5 = 0.034 \cdot (-0.16)$; $M5 = 0.034$; $IA5 = 0.035$.

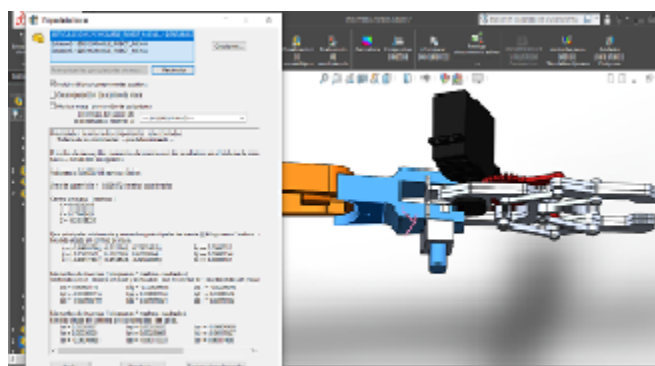


Figura 3.17: Articulación 5 y sus propiedades físicas.
Fuente: Diseño propio.

Esta sexta imagen permite visualizar los parámetros inerciales asociados a la sexta articulación:
 $XX6 = 0.0038$; $XY6 = 0.00030$; $XZ6 = -0.00049$; $YY6 = 0.0028$; $YZ6 = -0.0017$; $ZZ6 = 0.00114$; $MX6 = 0.100 \cdot (0.038)$; $MY6 = 0.100 \cdot (0.174)$; $MZ6 = 0.100 \cdot (-0.342)$; $M6 = 0.100$; $IA6 = 0.04$.

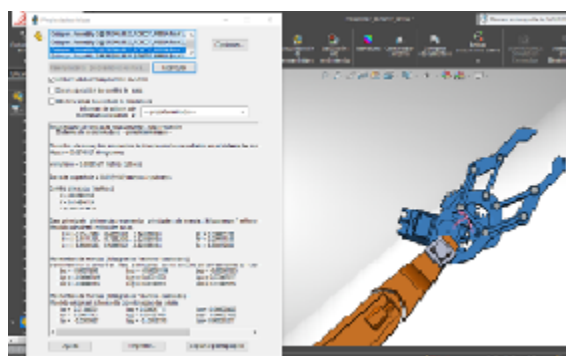


Figura 3.18: Articulación 6 y sus propiedades físicas.
Fuente: Diseño propio.

Teniendo en cuenta las propiedades físicas mostradas anteriormente de cada articulación se obtiene la tabla de parámetros 3.1.

J	XX	XY	YY	YZ	YZ	ZZ	MX	MY	MZ	M	Ia
1	XX1	XY1	XZ1	YY1	YZ1	ZZ1	MX1	MY1	MZ1	M1	IA1
2	XX2	XY2	XZ2	YY2	YZ2	ZZ2	MX2	MY2	MZ2	M2	IA2
3	XX3	XY3	XZ3	YY3	YZ3	ZZ3	MX3	MY3	MZ3	M3	IA3
4	XX4	XY4	XZ4	YY4	YZ4	ZZ4	MX4	MY4	MZ4	M4	IA4
5	XX5	XY5	XZ5	YY5	YZ5	ZZ5	MX5	MY5	MZ5	M5	IA5
6	XX6	XY6	XZ6	YY6	YZ6	ZZ6	MX6	MY6	MZ6	M6	IA6

Tabla 3.1: Tabla de de parámetros de un robot Puma

Teniendo en cuenta que los parámetros dinámicos de base permiten resumir los parámetros dinámicos a unos valores reagrupados que facilitan el cálculo obsérvese que en vez de tener 66 parámetros que definen la dinámica de este robot, con la aplicación de las fórmulas de los parámetros de base y teniendo en cuenta las consideraciones adicionales se obtienen solamente 19 [19].

La tabla de parámetros geométricos es introducida en un archivo .m en el software de Matlab y así es como se genera el modelo geométrico directo (MGD), el modelo dinámico directo (MDD) y el modelo dinámico inverso (MDI). Estos datos seran usados para realizar el control y simulación en Matlab/Simulink y serán llamados puma_directo.m y puma_inverso.m

A continuación en la tabla 3.2 se muestran los resultados.

j	XX	XY	XZ	YY	YZ	ZZ	MX	MY	MZ	M	Ia
1	0	0	0	0	0	0.070	0	0	0	0	0
2	-0.0152	0	0	0	0	0.0217	0.0821	0.0255	0	0	0
3	0.0129	0	0	0	0	0.0139	0	0	0	0	0.00018
4	0.0056	0	0	0	0	0.0058	0	0.0266	0	0	0.00035
5	0.0133	0	0	0	0	0.0131	0	0	0	0	0.00035
6	0.0030	0	0	0	0	0.0033	0	0	0	0	0.0004

Tabla 3.2: Parámetros de base de robot PUMA

3.3. Impresión de eslabones

La fabricación aditiva (AM de Additive Manufacturing), también conocida generalmente como impresión 3D, permite la fabricación de objetos totalmente personalizados con un alto nivel de complejidad geométrica y con una importante reducción del tiempo y del coste de fabricación[24].

La impresión 3D es un grupo de tecnologías de fabricación por adición capaz de crear un objeto tridimensional mediante la superposición de capas sucesivas de un determinado material. Por lo tanto, se trata de un proceso en el que se crea un objeto físico en tres dimensiones a través de un objeto o modelos digital mediante una impresora 3D, la cual puede usar diferentes tecnologías y

materiales para ir superponiendo capas hasta crear una réplica perfecta.

Para la elaboración de los eslabones se hizo uso del laboratorio de fabricación "FabLab" de la Universidad del Cauca que cuenta con dos impresoras 3D. Seguidamente el diseño del prototipo hecho en el software Solidworks fue exportado como un archivo *.stl* para luego abrir los archivos en el ambiente 3D de cada impresora, estas impresiones se realizaron en Ender 3D y Dremel 3D. Para la impresión de los primeros tres eslabones en Ender 3D una vez obtenido los archivos en *.stl* se realiz lo siguiente:

1. Se abre el software Ultimake Cura donde se modifican los parámetros como lo es el grosor de las líneas de impresión, el nivel de temperatura a usar, entre otros para realizar una correcta impresión.

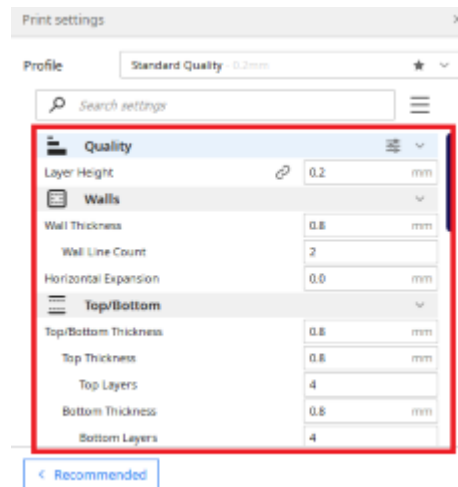


Figura 3.19: Parámetros en Utimake Cura.
Fuente: Diseño propio.

2. Una vez se exporta la pieza en el programa se verifica que la este correctamente ubicada en la cama de la impresora en el ambiente 3D, ya que exactamente así como se ve en el ambiente 3D se hará la impresión.

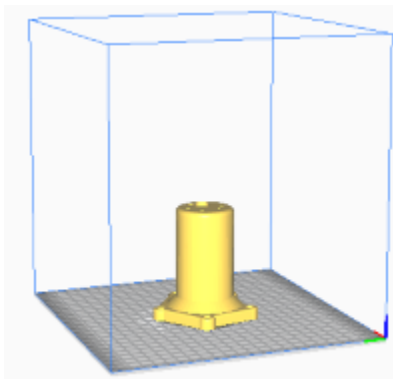


Figura 3.20: Eslabón 1. Escenario Ultimake Cura.
Fuente: Diseño propio.

3. A continuación el software calcula el tiempo de operación de la impresión y seguidamente se guarda en una memoria que sera introducida en la maquina 3D.

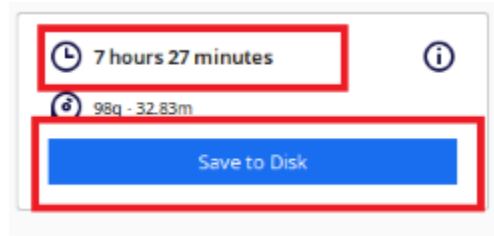


Figura 3.21: Cálculo de tiempo de impresión.

Fuente: Diseño propio

4. Finalmente se introduce en la impresora el material escogido que es el PLA (es el filamento por defecto elegido para la mayoría de las impresoras 3D basadas en extrusión porque puede imprimirse a baja temperatura y no requiere una cama calentada), se introduce la memoria y se abre el archivo en la maquina 3D e iniciamos con la opción start.



Figura 3.22: Impresora Ender 3D.

Fuente: Diseño propio

5. Una vez completado todas las capas hasta obtener la pieza final en el tiempo requerido se obtiene la pieza finalizada.



Figura 3.23: Eslabón impreso 1. Vista frontal
Fuente: Diseño propio.



Figura 3.24: Eslabón impreso 1. Vista superior
Fuente: Diseño propio.

Teniendo en cuenta el proceso descrito anteriormente se realizó el mismo procedimiento para el eslabón 2 y 3. Obteniendo los siguientes resultados:

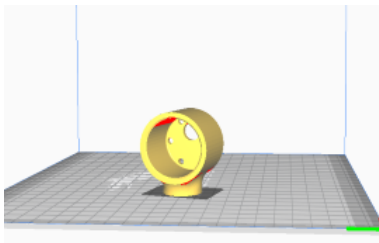


Figura 3.25: Eslabón 2. Escenario Ultimake Cura.
Fuente: Diseño propio.

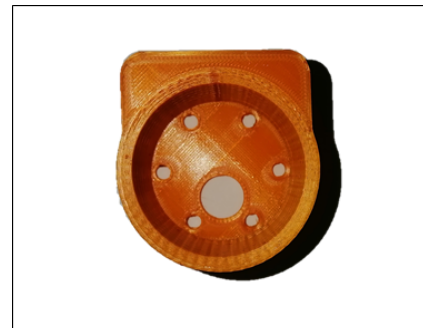


Figura 3.26: Eslabón impreso 2. Vista superior
Fuente: Diseño propio.

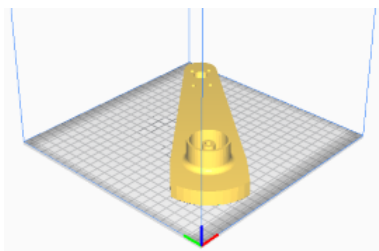


Figura 3.27: Eslabón 3. Escenario Ultimake Cura.
Fuente: Diseño propio.



Figura 3.28: Eslabón impreso 3. Vista superior
Fuente: Diseño propio.

Para los dos últimos eslabones se usó la impresora Dremel 3D. Esta es usada en proyectos académicos por su facilidad de uso, también ofrece buena calidad y robustez en las piezas de la siguiente manera obtenemos el eslabón 5 y 6 :

1. Para el primer paso se lleva a cabo el proceso de la misma manera que en la anterior impresora, se debe tener en cuenta que los archivos estén en *.stl* y de esta manera se abre el archivo en el escenario 3D de Dremel para la preparación y configuración de la pieza. En la parte izquierda del software se encuentran los parámetros a modificar para obtener una buena impresión y ubicar la pieza de manera que no se salga de las dimensiones de la cama.

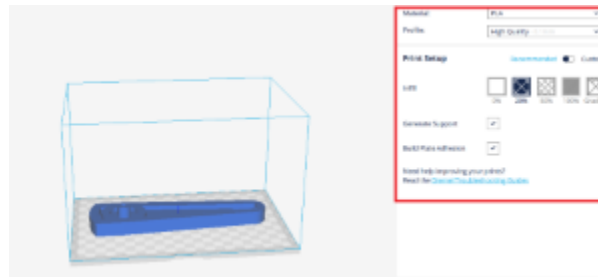


Figura 3.29: Eslabón 4. Escenario Dremel.

Fuente: Diseño propio.

2. Para el segundo paso se coloca y se prepara el filamento, se carga el archivo a la impresora por puerto USB y se inicia automáticamente la preparación de la impresión para la obtención de la pieza.

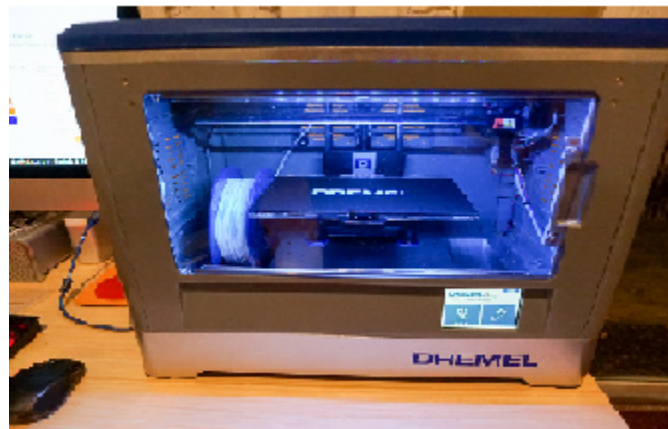


Figura 3.30: Impresora 3D Dremel.

Fuente: Diseño propio.

Las operaciones explicadas anteriormente también se realizan para la pieza 5 y finalmente se obtiene el resultado final.



Figura 3.31: Eslabón impreso 4. Vista superior
Fuente: Diseño propio.

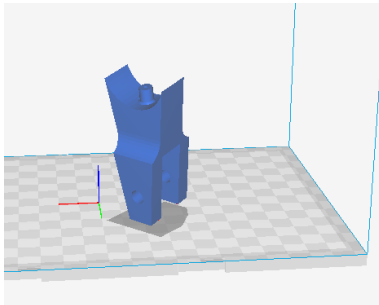


Figura 3.32: Eslabón 5 escenario Dremel.
Fuente: Diseño propio.

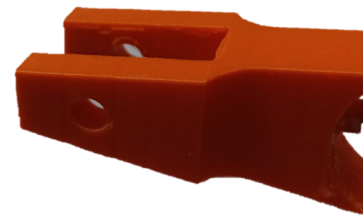


Figura 3.33: Eslabón impreso 5. Vista isométrica
Fuente: Diseño propio.

3.4. Elementos de medición y accionamiento

Se presenta los componentes elegidos para desempeñar acciones de medición de variables y ejecución de los esfuerzos de control, incluyendo las características y/o adaptaciones realizadas para la implementación en el robot.

3.4.1. Sensores

Para obtener una medición de la posición angular, es necesario usar un potenciómetro rotatorio con un rango aproximado de 250 grados con el objetivo de conocer la posición absoluta del eje del motor sin pérdida de referencia al momento de cortar la alimentación del sistema.

Para el caso del control CTC donde se hace necesario determinar la velocidad angular de cada una de las cadenas cinemáticas, se hace uso de calculo derivativo a través de software. Un encoder rotativo es un elemento que indica mediante posiciones codificadas su posición. El ky-040 es un encoder rotativo con 12 posiciones (cada 30°) e infinito, es decir, que podemos dar vueltas hacia ambos lados sin límite. Cuando pasamos por cada paso se nota un pequeño resalte que indica que se ha llegado a la nueva posición. Estos codificadores constan de dos pines para el pulsador (funciona como un pulsador normal) y tres pines para el codificador. Los tres pines del codificador van conectados uno a masa y los otros dos a las respectivas entradas que designemos en la placa Arduino.

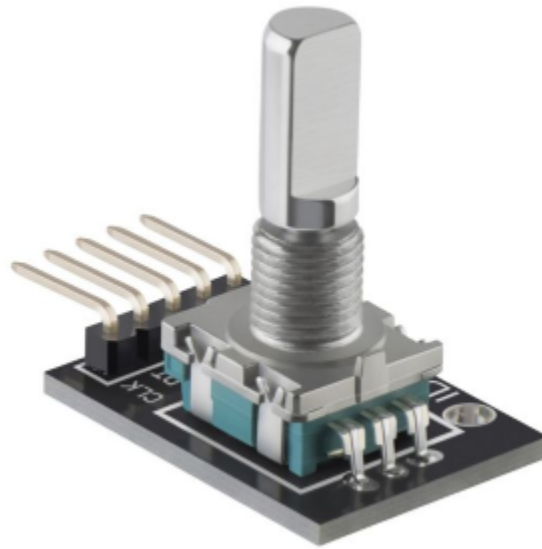


Figura 3.34: Potenciómetro Rotatorio Ky 040 incremental 360 Grados.
Fuente: Diseño propio

Características:

- Modelo: KY-040
- Tipo: Encoder incremental
- Ciclos por revolución (CPR): 20
- voltaje de funcionamiento: 0 - 5V
- Material: PCB
- Peso: 10g
- Tamaño: 32 x 19 x 30 mm

3.4.2. Actuadores

El diseño del robot trabajado requiere seis actuadores que presenten un movimiento rotacional. Para esto se decidió implementar los motor-reductores de corriente continua mostrados a continuación. Cuentan con un engranaje que permite disminuir la velocidad y al mismo tiempo incrementar el par proporcionado por el motor.

Este tipo de motor sera usado en el eslabón 1 y 2 (ver figura 3.24 y 3.26)

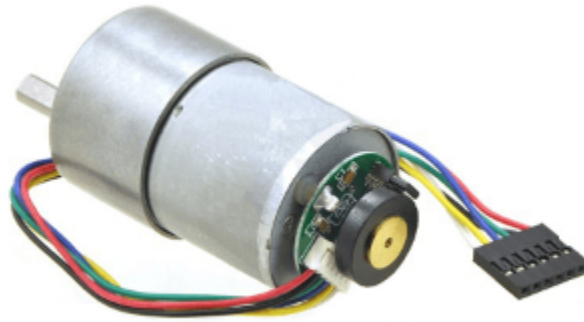


Figura 3.35: Motorreductor de 30:1 con Encoder POL1443.
Fuente: Diseño propio.

Características:

- Voltaje de alimentación: 12V DC.
- La salida de la caja reductora: 350 Rpm.
- La corriente de trabajo es de 300mA sin carga y de 5A a plena carga.
- Posee un torque de 8 Kg.cm.
- La resolución del encoder es de 64 cuentas por revolución (CPR), que corresponde a 1920 CPR del eje de salida de la caja de engranajes.
- Posee un conector de 6 cables, dos para la alimentación del motor y los otros 4 para la alimentación y las salidas de señal del encoder.
- Con 37 mm de diámetro, 64 mm de longitud, y pesa 210 g.

El motorreductor sera usado para el movimiento el eslabón 3 mostrado en la figura 3.28.



Figura 3.36: Motorreductor Pelv.
Fuente: Diseño propio.

- Motorreductor de 155 Rpm
- Fuente de voltaje: 12 Voltios
- Torque 4 kg.cm

Este motorreductor fue seleccionado para la rotación del eslabón 4 mostrado en la figura 3.3.



Figura 3.37: Motorreductor DC MR12120.
Fuente: Diseño propio.

Características:

- Motorreductor de 120 Rpm.
- Con alimentación de 12V DC a 30 mA.
- Posee un torque de 4 Kg.cm.
- El diámetro es de 20 mm.
- El diámetro del eje es de 4 mm.
- La longitud del eje es de 8 mm.

Por ultimo se escogió el servomotor para utilizar en el eslabón 5 mostrado en la figura 3.33.



Figura 3.38: Servomotor CDS5516.
Fuente: Diseño propio.

Características:

- Piñonería: Engranajes metálicos.
- Voltaje nominal: 6-16V.
- Torque 16 kg.cm.
- Interfases: Red serial TTL (7343bps 1Mbps) (soporta controlador AX12).
- Ángulo de operación: 300° .
- Resolución: 0.32° .

3.5. Elementos de control

El sistema de control esta compuesto por los componentes hardware encargados de procesar la información necesaria para alimentar los lazos de control y ejecutar las acciones determinadas.

3.5.1. Tarjeta electrónica de control

Se trabaja bajo la referencia Arduino MEGA 2560 (ver figura 3.5.1), se distingue por ser un microcontrolador bastante usado en ámbitos de estudio e investigación por su alta eficiencia para resolver proyectos de alta complejidad.

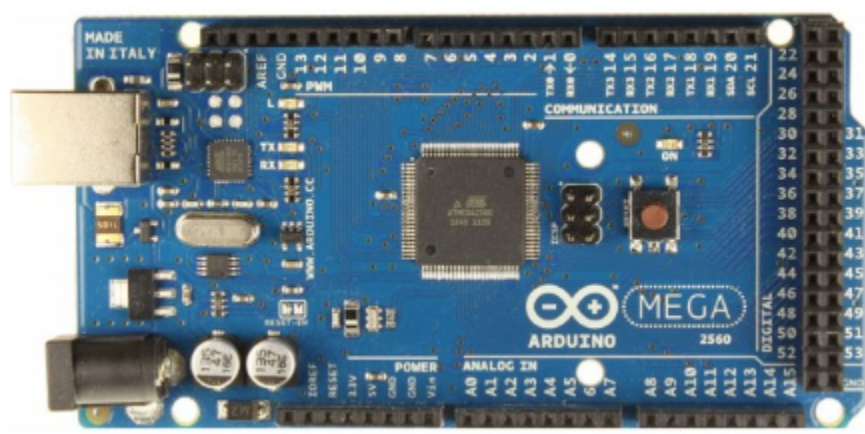


Figura 3.39: Arduino MEGA 2560.
Fuente tomada de [25].

Tiene 54 pines de entrada/salida digital (de los cuales 15 se pueden usar como salidas PWM), 16 entradas analógicas, 4 UART (puertos seriales de hardware), un oscilador de cristal de 16 MHz, una conexión USB, un conector de alimentación, un cabezal ICSP, y un botón de reinicio. Contiene todo lo necesario para soportar el microcontrolador. [25]. Entre otras características están :

- Velocidad del reloj: 16 MHz
- Voltaje de entrada de $-7-12V$.
- 256k de memoria flash.
- Velocidad del reloj de 16Mhz.
- Dimensiones: 102mm x 53mm

3.6. Elementos de sistema de potencia

En esta sección se describe los elementos encargados de la alimentación y distribución de potencia a todos los elementos del sistema electrónico del robot, permitiendo el suministro necesario para la correcta operación de todos los componentes electrónicos del prototipo.

3.6.1. Fuente de alimentación

Para la alimentación eléctrica del sistema se usó una fuente de alimentación regulada que brinda la potencia necesaria para el funcionamiento de los actuadores del sistema. Cumple con seguridad CE y RoHS, alta eficiencia y rendimiento estable. Es un gran transformador de potencia para equipos electrónicos, iluminación LED, electrodomésticos, etc.



Figura 3.40: Fuente de voltaje S-180-12.
Fuente: Diseño propio.

Características:

- Interruptor de fuente de alimentación, convertir CA 110 V a DC 12 V 15 A 180 W.
- Protección: protección contra escasez, protección contra sobrecarga, protección contra sobretensión.

3.6.2. Módulos de control de motores

El sistema de potencia es crucial para alimentar los motores debido a su mayor demanda energética y características propias de cargas inductivas, resultando necesario el uso de dos módulos puente H con referencia L298N para realizar el control de potencia y sentido de giro de los actuadores por medio de pulsos PWM y un driver puente H L9110s.

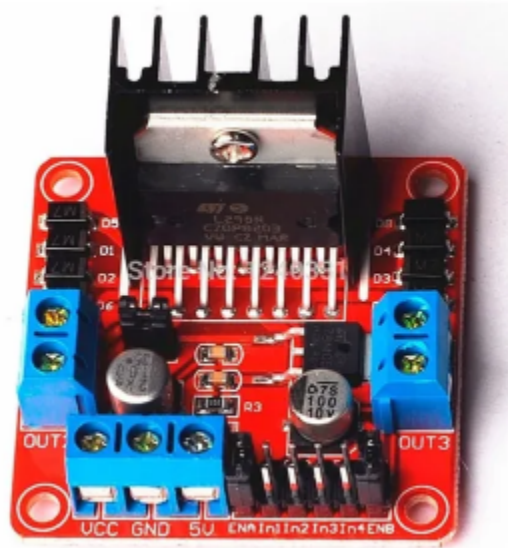


Figura 3.41: Puente H L298n
Fuente: Diseño propio.

Figura 3.42: Puente H L298n
Fuente: Diseño propio.

Características:

- Canales: 2 (soporta 2 motores DC o 1 motor paso a paso).
- Voltaje de control: 5V
- Voltaje de potencia (V motor): 5V - 35V DC.
- Consumo de corriente (lógico): 0 a 36mA.
- Capacidad de corriente: 2A (picos de hasta 3A).
- Potencia máxima: 25W.

3.7. Conexión eléctrica de componentes

Las conexiones eléctricas del sistema de control para el robot Puma se caracterizan principalmente por contener todas las conexiones cableadas entre la tarjeta controladora y los sensores; así como la conexión necesaria para la comunicación I2C. Como se observa la conexión se compone de elementos principales que son: los sensores de posición representados por los potenciómetros y el controlador Arduino Mega.

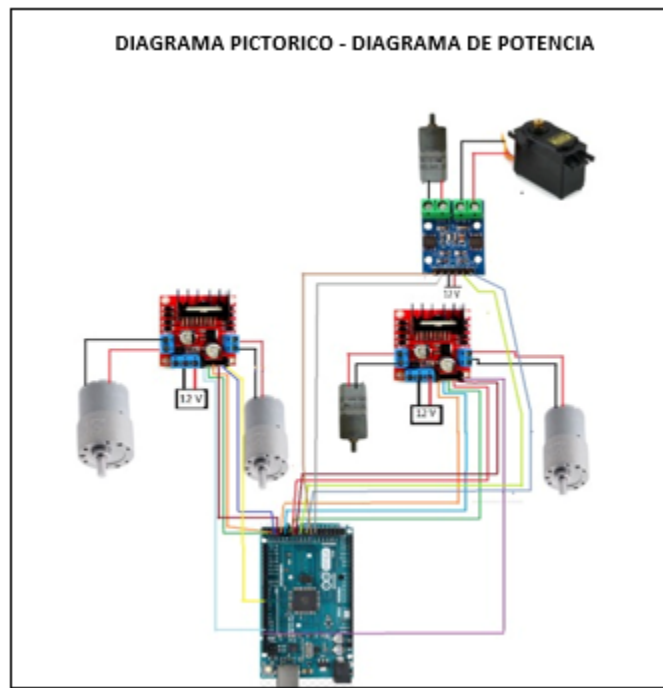


Figura 3.43: Conexión de Potencia

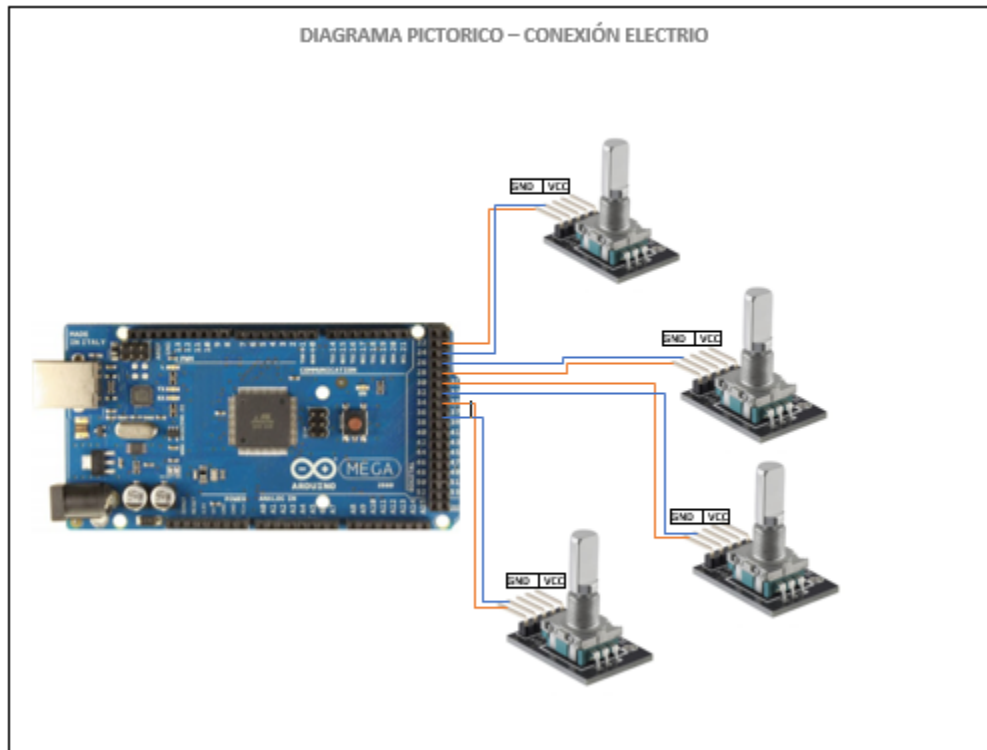


Figura 3.44: Conexión eléctrica

3.8. Análisis de la posición angular

El proceso de la toma de datos de los encoders asociados a cada motor se realizó a través de bloques realizados en Matlab/Simulink, a continuación se muestra el diagrama de bloques utilizado.

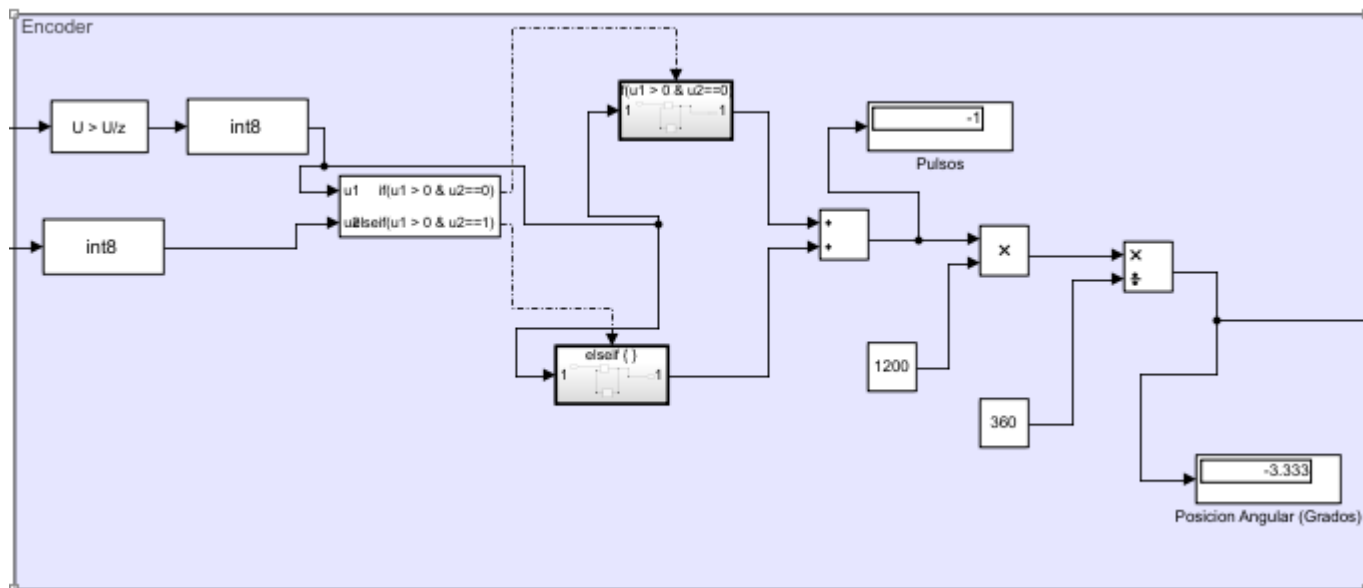


Figura 3.45: Diagrama de bloques de posición angular

Este diagrama consiste básicamente en la toma de datos a través de los pines digitales asignados, para luego ser procesados por un condicional **if** el cual va a asignar la posición concreta de cada motor, se observan también dos bloques de constantes las cuales son los valores que se deben asignar tanto para hacer la conversión de giro ($360/1$), como para determinar la cantidad de pulsos que otorga el encoder (1200).

3.9. Protocolos de comunicación

Los protocolos son necesarios para comunicar los distintos elementos electrónicos del sistema, permiten mantener un flujo de información desde los sensores hasta el sistema de control y desde sistema de control a los actuadores. Para establecer esta comunicación entre los distintos componentes del sistema de control se ha decidió utilizar dos protocolos de comunicación denominados I2C y USART.

3.9.1. Protocolo de comunicación UART

UART (universal asynchronous receiver / transmitter, por sus siglas en inglés), define un protocolo o un conjunto de normas para el intercambio de datos en serie entre dos dispositivos. UART es sumamente simple y utiliza solo dos hilos entre el transmisor y el receptor para transmitir y recibir en ambas direcciones. Ambos extremos tienen una conexión a masa. La comunicación en UART puede ser simplex (los datos se envían en una sola dirección), semidúplex (cada extremo se comunica, pero solo uno al mismo tiempo), o dúplex completo (ambos extremos pueden transmitir

simultáneamente). En UART, los datos se transmiten en forma de tramas. A continuación se describe y explica el formato y el contenido de estas tramas de forma sucinta.

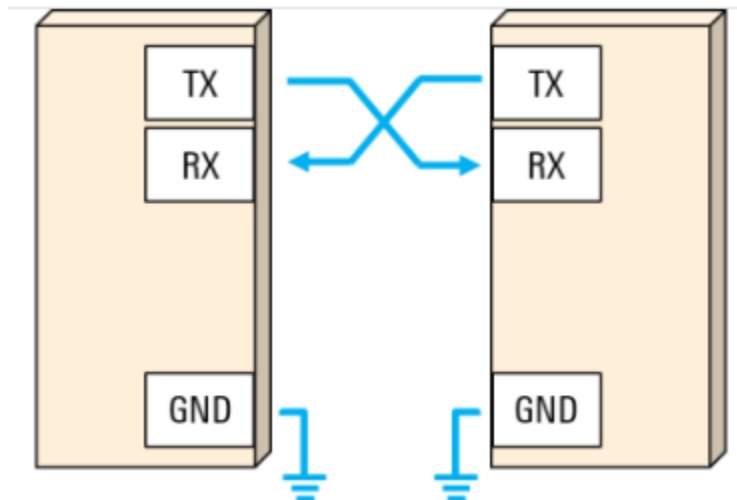


Figura 3.46: Esquema protocolo UART

Una de las mayores ventajas de UART es que es asíncrono: el transmisor y el receptor no comparten la misma señal de reloj. Si bien esto simplifica en gran medida el protocolo, plantea determinados requisitos en el transmisor y el receptor. Puesto que no comparten un reloj, ambos extremos deben transmitir a la misma velocidad, previamente concertada, con el fin de mantener la misma temporización de los bits [26].



Figura 3.47: Formato de trama en UART

Este protocolo es usado para comunicar el microcontrolador Arduino Mega 2560 con la computadora que contiene el algoritmo de control como se observa en el esquema la comunicación se realiza de forma automática al utilizar el paquete para Arduino suministrado por MathWorks. El algoritmo usado en Arduino para establecer la comunicación Simulink-Arduino se encuentra en la librería del paquete Arduino IO instalado en el programa Matlab, es un algoritmo que permite ajustar la board que vamos a utilizar que en este caso es la Arduino Mega 2560, este se compila

dentro de la tarjeta para su respectiva configuración y paso seguido la comunicación entre la placa y el programa Matlab queda ajustado.



Figura 3.48: Comunicación Arduino a Matlab/Simulink.

```

Archivo  Editar  Programa  Herramientas  Ayuda
adidos
case 212:
/* the fourth received value indicates the second pin
   from abs('c')=99, pin 2, to abs('!')=166, pin 69  */
if (val>98 && val<167) {
    pin=val-97;          /* calculate pin          */
    Enc[enc].pinB=pin;   /* set pin B          */

    /* set encoder pins as inputs          */
    pinMode(Enc[enc].pinA, INPUT);
    pinMode(Enc[enc].pinB, INPUT);

    /* turn on pullup resistors          */
    digitalWrite(Enc[enc].pinA, HIGH);
    digitalWrite(Enc[enc].pinB, HIGH); |

    /* attach interrupts          */
    switch(enc) {
        case 0:
            attachInterrupt(getIntNum(Enc[0].pinA), isrPinAEn0, CHANGE);
            attachInterrupt(getIntNum(Enc[0].pinB), isrPinBEn0, CHANGE);
            break;
        case 1:
            attachInterrupt(getIntNum(Enc[1].pinA), isrPinAEn1, CHANGE);
            attachInterrupt(getIntNum(Enc[1].pinB), isrPinBEn1, CHANGE);
            break;
        case 2:
            attachInterrupt(getIntNum(Enc[2].pinA), isrPinAEn2, CHANGE);
            attachInterrupt(getIntNum(Enc[2].pinB), isrPinBEn2, CHANGE);
            break;
    }
}
Subido

```

Figura 3.49: Código de configuración Arduino a Matlab/Simulink .

3.9.2. Configuración del software de control

Para establecer la comunicación entre Simulink y los microcontroladores, se usaron los bloques "Digital Read" encargados de leer los datos de posición entregados por los sensores de posición; y los bloques "Analog Write" para enviar hacia el drive L298N el esfuerzo de control necesario.

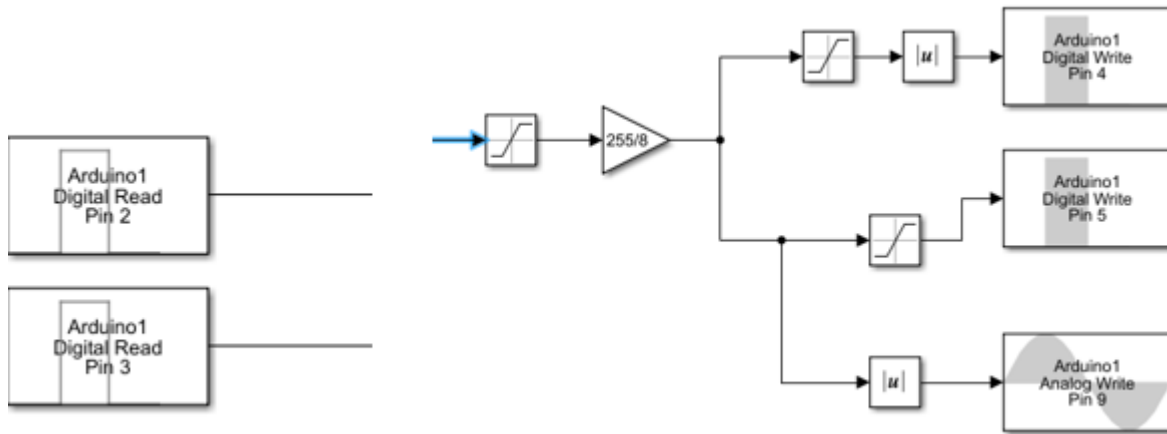


Figura 3.50: Bloques de lectura y escritura de datos.

Para configurar de manera adecuada la lectura de la posición enviada a través, se le asignaron al bloque "Digital read" los parámetros necesarios para realizar su debida configuración.

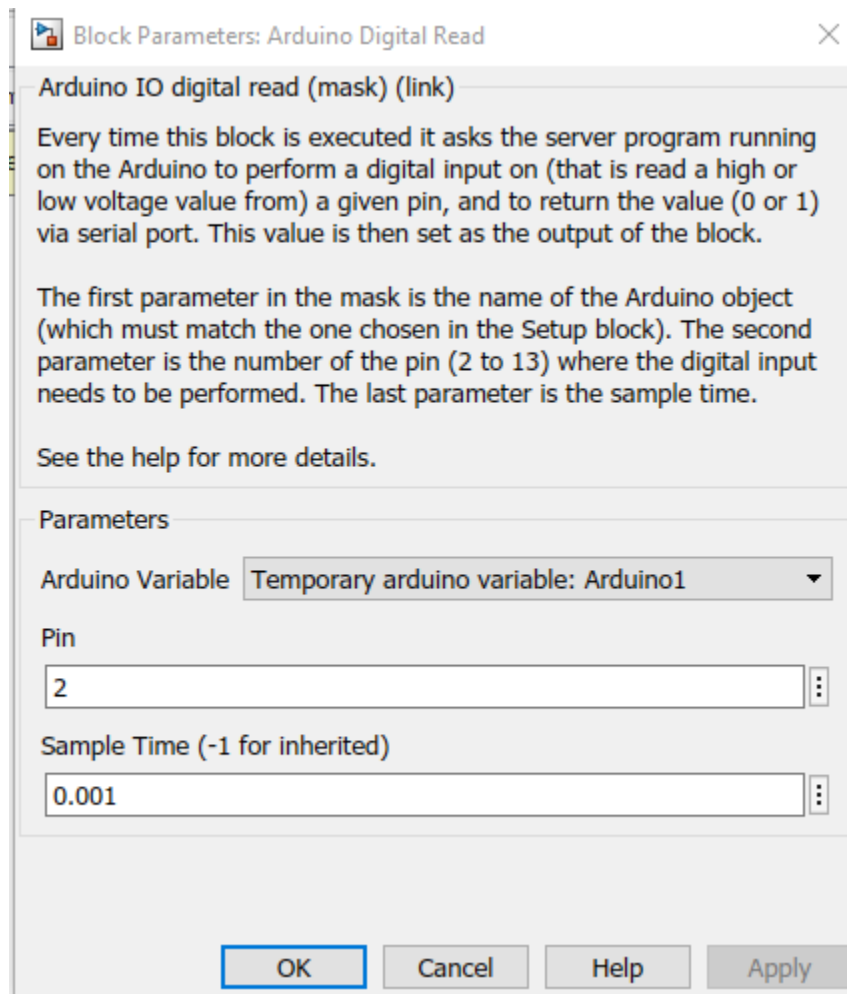


Figura 3.51: Configuración bloque Digital read.

De la misma forma se configuró el bloque de Digital Write para la parte de asignación del esfuerzo de control.

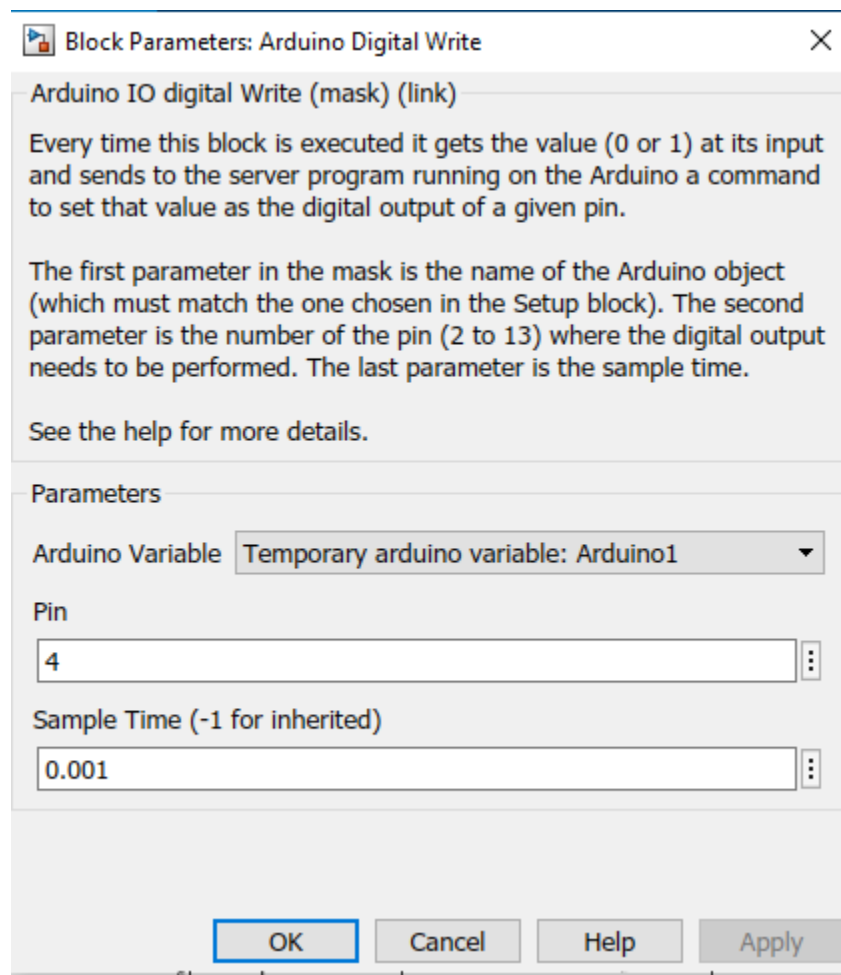


Figura 3.52: Configuración bloque Digital Write

3.9.3. Consideraciones finales sobre el prototipo

El prototipo fue desarrollado para proporcionar una plataforma escalable, versátil y asequible para trabajos de experimentación e investigación en torno a la robótica y específicamente con robots manipuladores. El resultado obtenido da respuesta a estos requerimientos, siendo capaz de poner a prueba diferentes estrategias de control y de adaptarse a cambios de geometra, estructura o función con un mínimo esfuerzo.

También es necesario señalar que al tratarse de una primera versión del robot, se identificaron varios puntos destacables para proponer caminos de actualización y mejora en el comportamiento del sistema en futuras versiones así como consideraciones que abarcan desde sistemas de control evaluados pero no seleccionados hasta aspectos mecánicos con incidencia en el comportamiento del robot.

3.9.4. Consideraciones del sistema de control

Durante el desarrollo del robot se consideraron distintas alternativas para poder implementar las estrategias de control donde destaca el uso de una tarjeta RaspberryPi para la implementación de las estrategias, encontrando limitantes en la capacidad de respuesta de dicha tarjeta ante el manejo de múltiples señales simultáneas. A futuro, se recomienda la implementación de tarjetas

de adquisición de datos dedicadas para posibilitar la mejora de la instrumentación sin sacrificar los tiempos de respuesta del sistema.

3.9.5. Consideraciones sobre los sensores

En este apartado se encuentran dos consideraciones importantes, primeramente los sensores implementados representan un sistema sencillo y eficiente para la medición de las posiciones angulares, sin embargo en futuras versiones es aconsejable actualizar estos elementos a sistemas de mayor linealidad y mayor precisión sin necesidad de procesos de calibración y linealización de la medida. Por otra parte se recomienda la implementación de nuevos sistemas de retroalimentación de los lazos de control, especialmente en estrategias de medición cartesiana sobre el órgano terminal del robot, ya sean sistemas de medición absoluta como sistemas de visión maquina o sistemas de medición relativos a través del uso de sensores inerciales.

3.9.6. Consideraciones sobre el modelo 3D

El modelo 3D desarrollado representa con alta precisión y fiabilidad el comportamiento del sistema, todas las piezas que inciden a nivel geométrico y dinámico en los movimientos del prototipo están modeladas con fidelidad y pueden adaptarse a cambios de disposición para complementar el estudio del sistema desarrollado en futuros trabajos.

Capítulo 4

Modelo del robot

4.1. Análisis matemático y diseño de controladores

4.1.1. Descripción de la estructura y modelo matemático

El robot Puma es un brazo con seis articulaciones que le proporcionan seis grados de libertad y le permiten posicionar y orientar su herramienta final. Una articulación puede ser:

- Lineal (deslizante, traslacional o prismática), si un eslabón desliza sobre un eje solidario al eslabón anterior.
- Rotacional, en caso de que un eslabón gire en torno a un eje solidario al eslabón anterior.

Para este caso todas las articulaciones del robot son tipo rotoide. Teniendo en cuenta lo anterior el esquema del robot a construir en este proyecto es el siguiente:

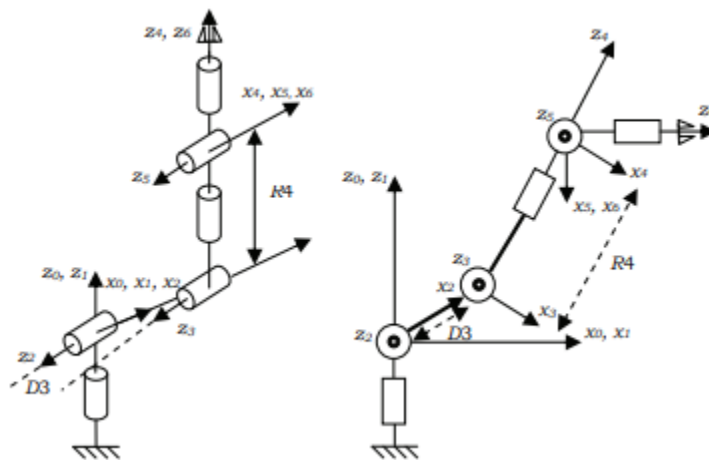


Figura 4.1: Arquitectura robot Puma.

Fuente: Tomada de [19].

En esta figura tomada de [19] se puede observar que las tres primeras articulaciones corresponden a un portador antropomórfico mientras que las tres últimas corresponden a una muñeca tipo rotula. La figura es mostrada de esta manera para poder realizar el análisis geométrico que se describirán en los siguientes ítems. Así mismo teniendo en cuenta lo descrito en el capítulo 3; se obtiene la

tabla descrita en 4.1 donde se analizan varios cuerpos unidos y movidos por cierto número de articulaciones, un sistema de coordenadas estará presente en cada articulación.

Para el análisis matemático es de vital importancia en cada tiempo de muestreo saber exactamente cuál es la posición y orientación del órgano terminal respecto a la base y qué articulaciones se deben mover para posicionar el órgano terminal en un punto deseado. Por ello son necesarios dos modelos para determinar dónde se encuentra el órgano terminal.

En el primer caso se habla del órgano terminal a la base (modelo geométrico directo) y en el segundo de la base al órgano terminal (modelo geométrico inverso).

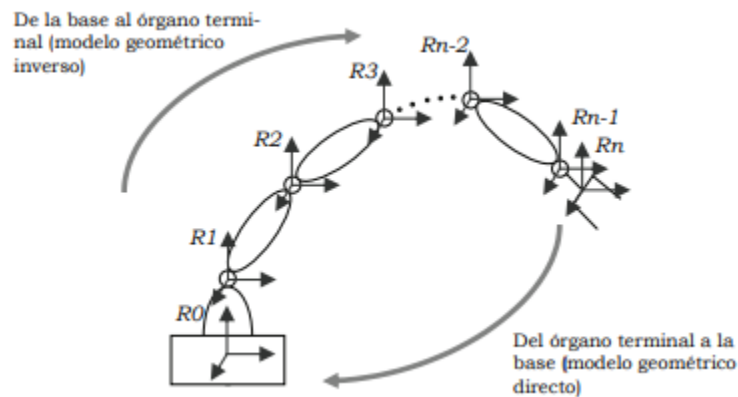


Figura 4.2: Sistema de coordenadas para articulación.

Fuente: Tomada de [19].

4.1.2. Modelos Geométricos

Para obtener los modelos geométricos del robot, es necesario encontrar una serie de parámetros para cada articulación, que dependen de los ángulos y las distancias entre los ejes de las articulaciones del robot PUMA, los cuales se muestra en la figura 4.1.1.

En la imagen (ver figura 3.12) se observa la arquitectura de un robot tipo Puma de seis grados de libertad donde observamos que todas sus articulaciones son de tipo rotoide y los valores que se asignan a la tabla depende del análisis, donde:

j : Es la articulación j .

σ_j : 0 si la articulación es rotoide, 1 si es prismática.

α_j : Ángulo entre los ejes z_{j-1} y z_j correspondiente a una rotación alrededor del eje x_j .

d_j : Distancia entre los ejes z_{j-1} y z_j a lo largo del eje x_j .

θ_j : Ángulo entre los ejes x_{j-1} y x_j correspondiente a una rotación alrededor del eje z_j .

r_j : Distancia entre x_{j-1} y x_j a lo largo de z_j .

Tras analizar las características del robot Puma, se obtiene la respectiva tabla de parámetros.

J	σ_j	α_j	d_j	θ_j	r_j
1	0	0	0	θ_1	0
2	0	90	0	θ_2	0
3	0	0	D3	θ_3	0
4	0	-90	0	θ_4	R4
5	0	90	0	θ_5	0
6	0	-90	0	θ_6	0

Tabla 4.1: Tabla de parámetros geométricos del robot PUMA

4.1.3. Modelo geométrico directo

Este modelo permite conocer la posición cartesiana y la orientación del órgano terminal. Para hallar el modelo geométrico directo (0T_n) es necesario aplicar la matriz, vista en [19]; a cada una de las articulaciones del robot Puma, utilizando para ello la tabla de parámetros geométrico (4.1). El modelo geométrico directo de este robot de seis grados de libertad estará representado por la matriz 0T_6 . A su vez esta matriz es igual a la multiplicación sucesiva de ${}^0T_6 = {}^0T_1 T_1^2 T_2^3 T_3^4 T_4^5 T_5^6$. Esto significa que deben hallarse cada una de las seis matrices que representan la transformación entre cada sistema de coordenadas de este robot. Para la primer articulación del robot Puma los valores geométricos son:

J	σ_j	α_j	d_j	j	r_j
1	0	0	0	θ_1	0

Reemplazando estos valores en la matriz se tiene:

$${}^0T_j = \begin{bmatrix} C\theta_j & -S\theta_j & 0 & d_j \\ C\theta_j S\alpha_j & C\alpha_j C\theta_j & -S\alpha_j & -r_j S\alpha_j \\ S\alpha_j S\theta_j & S\alpha_j C\theta_j & C\alpha_j & -r_j C\alpha_j \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^0T_j = \begin{bmatrix} \text{Cos}\theta_1 & -\text{Sen}\theta_1 & 0 & d_1 \\ \text{Cos}\theta_1 \text{Sen}\alpha_1 & \text{Cos}\alpha_1 \text{Cos}\theta_1 & -\text{Sen}\alpha_1 & -r_1 \text{Sen}\alpha_1 \\ \text{Sen}\alpha_1 \text{Sen}\theta_1 & \text{Sen}\alpha_1 \text{Cos}\theta_1 & \text{Cos}\alpha_1 & -r_1 \text{Cos}\alpha_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Se reemplaza entonces $\alpha_1 = 0, d_1 = 0, r_1 = 0$ en la ecuación anterior. De la misma manera, y utilizando los demás datos de la tabla de parámetros geométricos, se hallan las demás matrices de transformación, las cuales se detallan a continuación:

$${}^0T_1 = \begin{bmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \begin{bmatrix} C2 & -S2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S2 & C2 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \begin{bmatrix} C3 & -S3 & 0 & D3 \\ S3 & C3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3T_4 = \begin{bmatrix} C4 & -S4 & 0 & 0 \\ 0 & 0 & 1 & R4 \\ -S4 & -C4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4T_5 = \begin{bmatrix} C5 & -S5 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S5 & C5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^5T_6 = \begin{bmatrix} C6 & -S6 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ S6 & -C6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Con el fin de aligerar cálculos, se debe aplicar las siguientes formulas de sumas de senos y cosenos:

$$\text{sen}(a+b)=\text{cos}(a)\text{cos}(b)+\text{sen}(b)\text{cos}(a) \quad \text{cos}(a+b)=\text{cos}(a)\text{cos}(b)-\text{sen}(a)\text{sen}(b) \quad (4.1)$$

Para el Puma las articulaciones 2 y 3 son paralelas. La matriz resultante de multiplicar sus respectivas matrices de transformación es, aplicando las relaciones mostradas en 4.1 se simplifica a:

$${}^1T_3 = \begin{bmatrix} C23 & -S2 & 0 & C2D \\ 0 & 0 & -1 & 0 \\ S23 & C23 & 0 & S3D3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Se calcula ahora el modelo geométrico directo para el robot Puma, partiendo de la ultima articulación:

$$U_5 = {}^5T_6$$

$$U_4 = {}^4T_6 = {}^4T_5 {}^5T_6 = \begin{bmatrix} C5C6 & -C5S6 & -S5 & 0 \\ S6 & C6 & 0 & 0 \\ S5C6 & -S5S6 & C5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$U_3 = {}^3T_6 = {}^3T_4 U_4 = \begin{bmatrix} C4C5C6 - S4S6 & -C4C5S6 - S4C6 & -C4S5 & 0 \\ S5S6 & -S5S6 & C5 & R4 \\ -S4C5C6 - C4S6 & S4C5S6 - C4C6 & S4S5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Los vectores de la matriz U_2 son:

$$s_x = C3(C4C5C6 - S4S6) - S3S5C6$$

$$s_y = S3(C4C5C6 - S4S6) + C3S5C6$$

$$s_z = -S4C5C6 - C4S6$$

$$n_x = -C3(C4C5S6 + S4C6) + S3S5S6$$

$$n_y = -S3(C4C5S6 + S4C6) - C3S5S6$$

$$n_z = S4C5S6 - C4C6$$

$$a_x = -C3C4S5 - S3C5$$

$$a_y = -S3C4S5 + C3C5$$

$$a_z = S4S5$$

$$P_x = -S3R4 + D3$$

$$P_y = C3R4$$

$$P_z = 0$$

$$U_1 = {}^1 T_6 = {}^1 T_2 U_2 = {}^1 T_3 U_3$$

Los vectores de U_1 son:

$$\begin{aligned} s_x &= C23(C4C5C6 - S4S6) - S23S5C6 \\ s_y &= S4C5C6 + C4S6 \\ s_z &= S23(C4C5C6 - S4S6) + C23S5C6 \\ n_x &= -C23(C4C5S6 + S4C6) + S23S5S6 \\ n_y &= -S4C5S6 + C4C6 \\ n_z &= -S23(C4C5S6 + S4C6) - C23S5S6 \\ a_x &= -C23C4S5 - S23C5 \\ a_y &= -S4S5 \\ a_z &= -S23C4S5 + C23C5 \\ P_x &= -S23R4 + C2D3 \\ P_y &= 0 \\ P_z &= C23R4 + S2D3 \end{aligned}$$

Finalmente los vectores de la matriz U_0 son:

$$\begin{aligned} s_x &= C1(C23(C4C5C6 - S4S6) - S23S5C6) - S1(S4C5C6 + C4S6) \\ s_y &= S1(C23(C4C5C6 - S4S6) - S23S5C6) + C1(S4C5C6 + C4S6) \\ s_z &= S23(C4C5C6 - S4S6) + C23S5C6 \\ n_x &= C1(-C23(C4C5S6 + S4C6) + S23S5S6) + S1(S4C5S6 - C4C6) \\ n_y &= S1(-C23(C4C5S6 + S4C6) + S23S5S6) - C1(S4C5S6 - C4C6) \\ n_z &= -S23(C4C5S6 + S4C6) - C23S5S6 \\ a_x &= -C1(C23C4S5 + S23C5) + S1S4S5 \\ a_y &= -S1(C23C4S5 + S23C5) - C1S4S5 \\ a_z &= -S23C4S5 + C23C5 \\ P_x &= -C1(S23R4 - C2D3) \\ P_y &= -S1(S23R4 - C2D3) \\ P_z &= C23R4 + S2D3 \end{aligned}$$

En el libro [19] se encuentra que para hallar la matriz 0T_6 se puede proceder a multiplicar de derecha a izquierda cada una de las matrices (de derecha a izquierda: Se multiplica primero la matriz 4T_5 por 5T_6 para obtener la matriz 4T_6 . Luego se multiplica ésta por 3T_4 para obtener la matriz 3T_6 , y así sucesivamente).

Los dos métodos generan la misma matriz 0T_6 , pero se debe utilizar el segundo método (de derecha a izquierda), ya que los resultados parciales de este procedimiento son utilizados en el cálculo del modelo geométrico inverso, el cual se verá en la sección siguiente.

4.1.4. Modelo geométrico inverso

Para hallar los valores de las variables articulares se utiliza el método de Paul para las tres primeras $(\theta_1, \theta_2, \theta_3)$, y el caso particular de la muñeca tipo rótula para las tres últimas $(\theta_4, \theta_5, \theta_6)$.

a) Cálculo de $\theta_1, \theta_2, \theta_3$:

De acuerdo a la metodología de Paul se multiplica a la izquierda de la ecuación del modelo geométrico directo por 1T_0 :

$${}^1T_0 U_0 = {}^1T_6$$

La parte izquierda es:

$$U(1) = C1P_x + S1P_y$$

$$U(2) = -S1P_x + C1P_y$$

$$U(3) = P_z$$

La parte derecha está dada por la cuarta columna de 1T_6 :

$$T(1) = -S23R4 + C2D3$$

$$T(2) = 0$$

$$T(3) = C23R4 + S2D3$$

Igualando U(2) con T(2) se encuentran solución siguientes para θ_1 :

$$\theta_1 = \text{atan}(P_y, P_x)$$

$$\theta_1' = 1 + 180$$

La solución θ_1' muestra que una rotación de 180^0 , permitiendo alcanzar la posición U_0 deseada. Esto sucede solo en ciertos ángulos, requiriéndose un análisis cuidadoso para determinar si dicho movimiento adicional también lleva a U_0 . Siguiendo la solución descrita en [19], se tiene la solución para θ_2 y θ_3 :

$$\theta_2 = \arctan(S2, C2)$$

$$\theta_3 = \arctan(S3, C3)$$

Para el cálculo de $\theta_4, \theta_5, \theta_6$. En este último caso solo interesan solo las ecuaciones de orientación, de la misma manera se encuentra la solución en [19]

$$\theta_4 = \arctan(Hz, -Hx)$$

$$\theta_4 = \theta_4 + 180^0$$

$$\theta_5 = \arctan(S5, C5)$$

$$\theta_6 = \arctan(S6, C6)$$

$$\theta_6 = \theta_6 + 180^0$$

4.2. Diseño de controladores

Para la simulación del robot Puma se tuvo en cuenta los parámetros descritos en el capítulo 3 y también lo expuesto anteriormente ya que sin estos datos no se podría realizar una simulación y sería tedioso poder implementar los controladores descritos a continuación. La sección se divide en dos partes, en la primera se explica y se simula la parte del control por par calculado y en la segunda parte de igual manera es explicado y simulado el controlador PID.

4.2.1. Control por par calculado

El controlador por par calculado CTC es un controlador no lineal de gran alcance que se utiliza ampliamente en el control de robots manipuladores cuando estos requieren una evolución rápida y una gran precisión, para ello existe esta ley de control sofisticada que cumple los requerimientos correctamente.

Este tipo de control exige el cálculo del modelo dinámico en línea y el conocimiento de los valores numéricos de los parámetros dinámicos. Es decir el control lleva implícito el modelo del sistema que se quiere calcular, y por lo tanto es imprescindible conocer con cierta exactitud el valor de los parámetros dinámicos del robot. Este controlador funciona muy bien cuando se conocen todos los parámetros dinámicos y físicos, pero una desventaja es cuando los parámetros dinámicos tienen variaciones el controlador no tiene un rendimiento aceptable en el seguimiento de las trayectorias.

4.2.2. Esquema general del controlador CTC

El esquema que describe el control Par-calculado se evidencia en la figura 4.4, como se observa el control CTC cuenta con términos de control tipo PD, no obstante este controlador no es de tipo lineal ya que las ganancias de posición y velocidad dependen del error de posición. La parte punteada se refiere al modelo dinámico inverso (MDI) o modelo matemático de la planta (en este caso el robot).

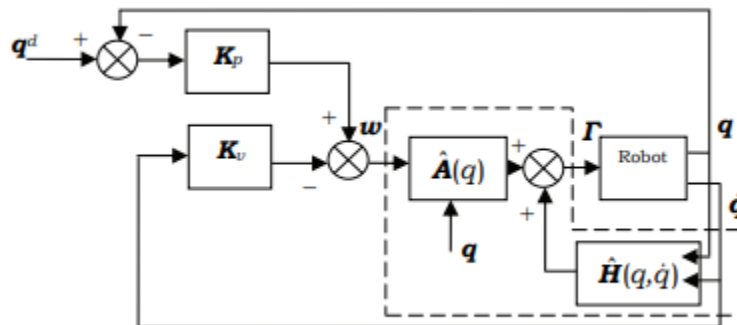


Figura 4.3: Esquema general control CTC.

Fuente: Tomada de [19]

4.2.3. Simulación en MATLAB - Simulink

Para realizar los dos tipos de control (tanto CTC como PID) se tiene en cuenta lo siguiente:

- Consignas: La consigna a seguir se define en Simulink con los bloques From Workspace. Es necesario cargar estas consignas antes de realizar la simulación.
- Modelo matemático del robot: Está representado por el modelo dinámico directo del robot, el cual se define en un archivo y es llamado por la S-Function. En este caso es llamado *puma_directo.m*
- Controlador: El controlador CTC cartesiano que involucra el modelo del robot se define por la MATLAB fcn. El archivo representará el modelo dinámico inverso del robot en este caso es llamado *puma_inverso.m*

- Error cartesiano o articular: Es la herramienta que permite alcanzar los requerimientos del problema planteado. Estos errores se observan en un osciloscopio donde se analiza la diferencia entre las señal deseada y la señal que se obtiene. Estas señales son articulares para el caso del error articular y cartesianas para el caso del error cartesiano. Lo que nos indica el tamaño del error en la sintonización.

Los bloques que se muestran a continuación es lo descrito anteriormente

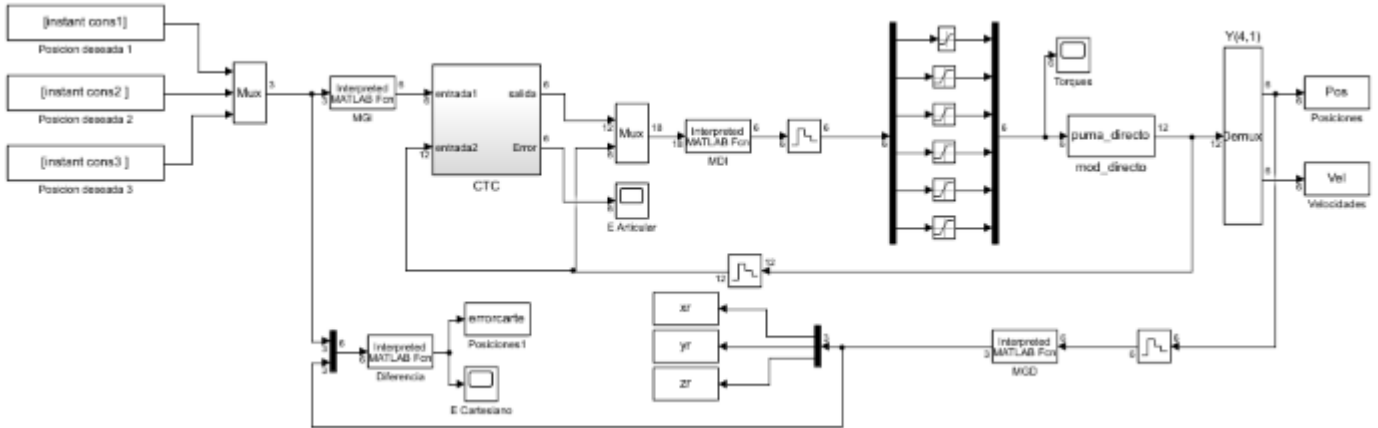


Figura 4.4: Esquema control CTC cartesiano.

Fuente: Tomada de [19]

El bloque llamado "CTC" se muestra en la figura 4.6 que contiene el control PD, un controlador PD (proporcional-diferencial) es un elemento de transferencia de un sistema de control de bucle cerrado que comprende componentes de elemento tanto P como D. El componente diferencial responde a la velocidad en la que el error de control cambia. El valor es multiplicado por el coeficiente de acción-derivada KD y sumado al componente P (lo que, por su parte, actúa proporcionalmente en un error de control específico). Como resultado, el controlador PD puede responder a un error de control inminente y, por lo tanto, lograr una acción derivada durante el proceso de control.

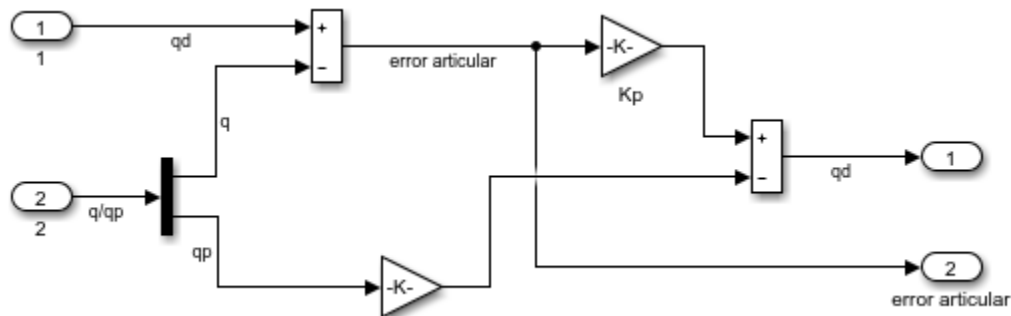


Figura 4.5: Control CTC para robot Puma.

Fuente: Modificada de [19]

Para lograr la sintonización del control CTC, se realizó el proceso mediante prueba y error, ajustando los valores de ganancia proporcional y ganancia derivativa, para así obtener un error demasiado pequeño (10^{-4}).

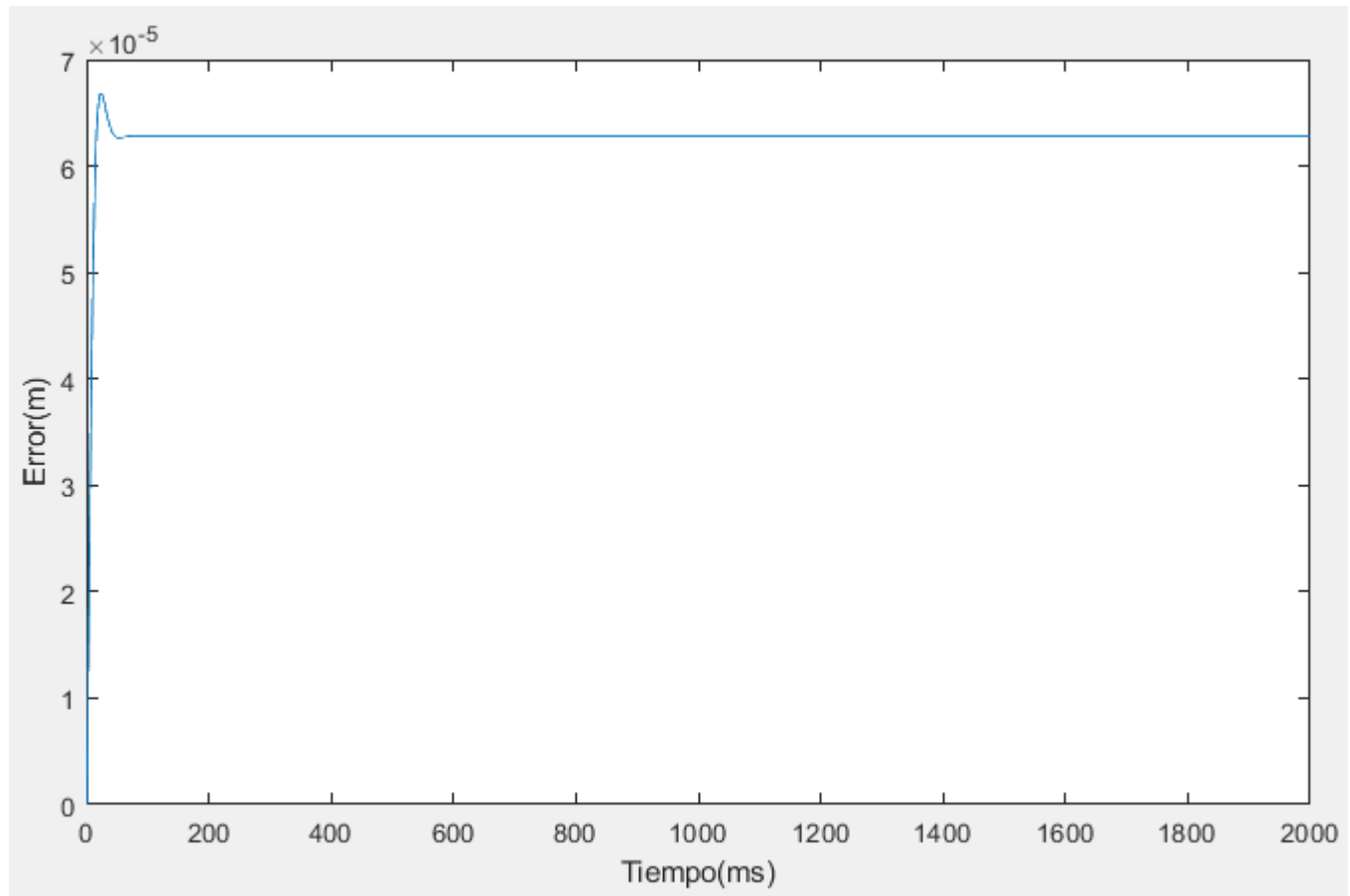


Figura 4.6: Error cartésiano del control CTC para un robot Puma.

Fuente: Diseño propio.

Como resultado de estas pruebas se obtiene la figura 4.7, donde se observa como la consigna obtenida sigue la consigna deseada.

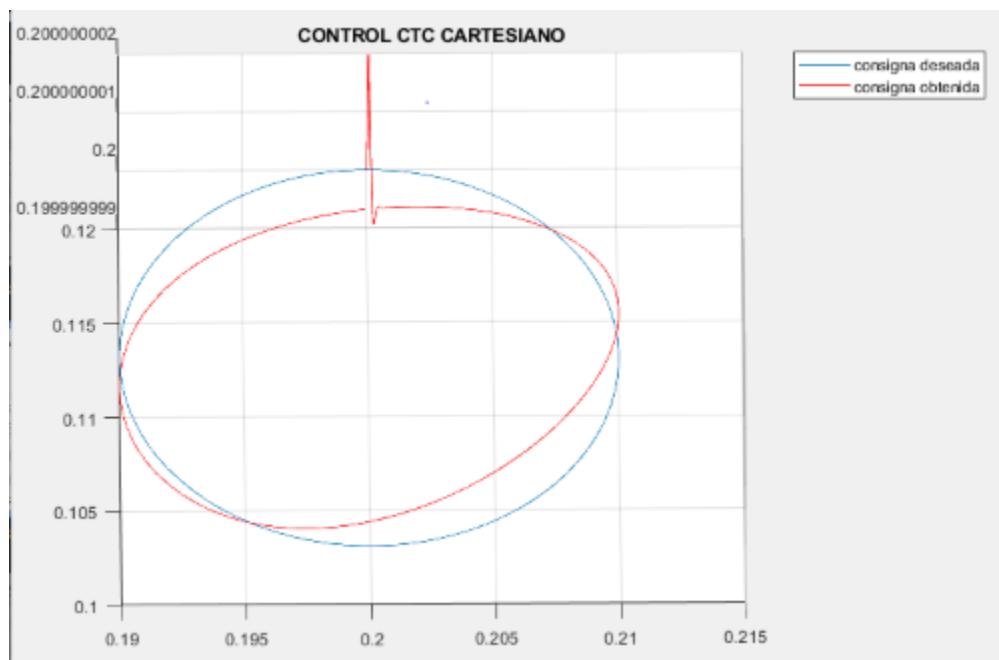


Figura 4.7: Trayectoria deseada vs trayectoria obtenida.

Fuente: Fuente propia

Para el esquema del control CTC articular la consigna deseada está definida en el archivo grado_cinco.m, también involucra el modelo de la planta representado por el modelo dinámico inverso (archivo puma_inverso.m). El esquema de esta estrategia es el siguiente:

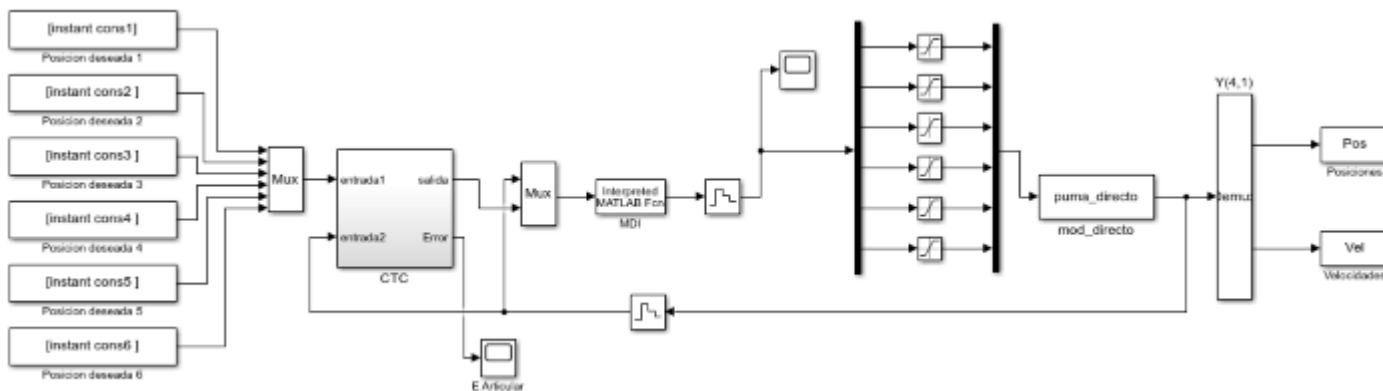


Figura 4.8: Esquema de control CTC articular.

Fuente: Fuente propia

Para el subsistema del control se tiene lo descrito en el caso del control cartesiano, un control PD, mostrado a continuación:

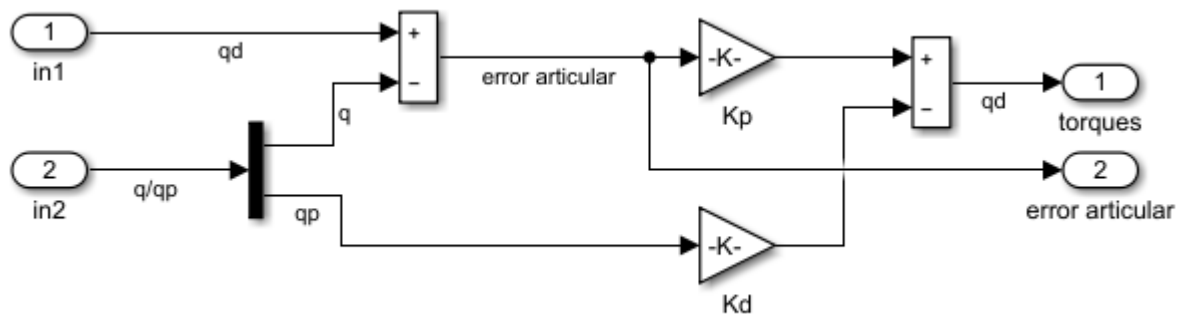


Figura 4.9: Control CTC para robot Puma.

Fuente: Modificada de [19]

De igual manera que en el caso anterior los resultados de la sintonización se obtuvieron a prueba y error obteniendo un error demasiado pequeño (10^{-5}). A continuación se entrega la diferencia entre las señales deseadas y las medidas

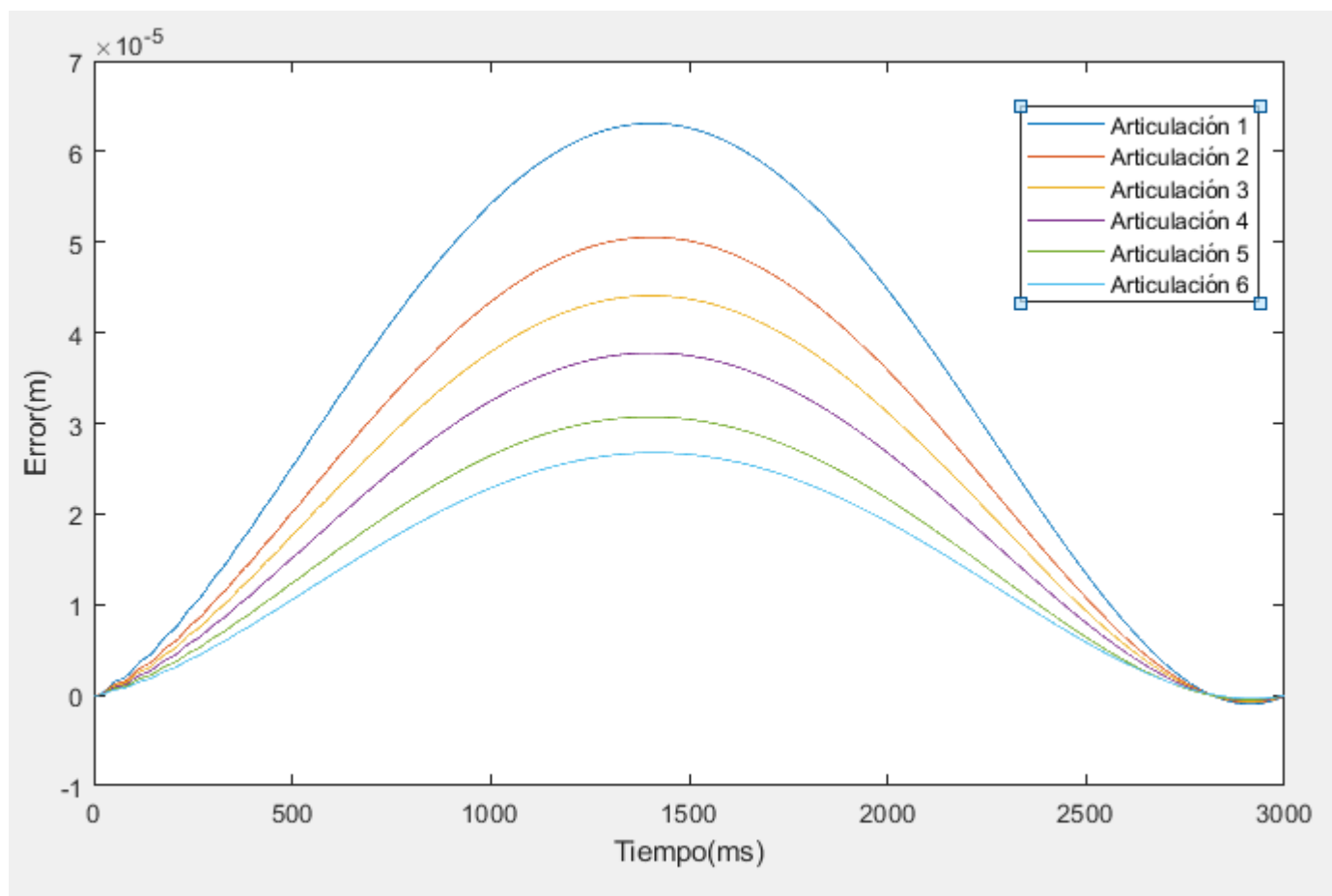


Figura 4.10: Error articular del control CTC para un robot Puma.

4.2.4. Control PID

Los controladores PID son los más utilizados en toda clase de industrias, y ofrecen unas prestaciones muy aptas para la mayoría de sistemas de control. Una de sus ventajas es la capacidad de ofrecer un efectivo cumplimiento de la tarea de control aun desconociendo el modelo matemático [27]. Pero en contraste se pueden encontrar malas precisiones y desplazamientos excesivos en el caso de movimientos rápidos [19].

El control PID se caracteriza por contar con tres términos de ganancia: proporcional, encargada de disminuir el error entre un estado deseado y el estado actual del sistema con la desventaja de crear oscilaciones; integral, que mediante la suma de áreas bajo la curva del error de posición disminuye el error de estado estacionario; y derivativa, que reduce las oscilaciones y cambios bruscos en la acción de control del sistema, logrando en conjunto una acción de control completa [28]. Sintonizar controladores de tipo Proporcional + Integral + Derivativo (PID), no es trivial, y a pesar de que existen diversos estudios para sintonizar estos controladores, parece no haberse resuelto el problema para tener una técnica que pueda ser implementada en los diversos sistemas. Los sintonizadores clásicos están limitados a sistemas de una entrada y una salida (SISO), de naturaleza lineal y que son estables en lazo abierto o los métodos inteligentes como las redes neuronales están limitados a los recursos computacionales [29]. Aunque los controladores PID son ampliamente usados en los procesos industriales, su efectividad es frecuentemente limitada debido a una sintonización pobre. La sintonización manual de las ganancias de un controlador es una tarea que consume demasiado tiempo [30], no es una sintonización óptima y entre más articulaciones tenga más difícil se vuelve.

Los estudios para obtener un buen resultado de control con el PID ha evolucionado considerablemente mostrando diferentes técnicas basadas en el control difuso. La aplicación de estas técnicas a sistemas reales es cada vez una tendencia con mejores posibilidades de control, ofreciendo resultados en un menor tiempo de respuesta comparado a un clásico controlador PID, como lo demuestran en [31].

4.2.5. Esquema general del controlador PID

El control PID se puede esquematizar de la siguiente forma, siendo $G_p(s)$ el modelo matemático de la planta a controlar, y $G_c(s)$ la función de transferencia del controlador PID, dividido en sus tres acciones básicas: proporcional (K_p), integral (K_i) y derivativa (K_d) [27]. Su esquema general se encuentra en la figura 4.11

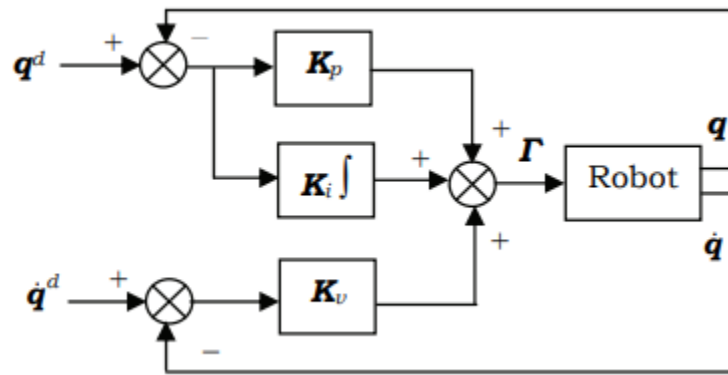


Figura 4.11: Esquema general control PID.

Fuente: Tomada de [19]

Observamos que en el bloque *Robot* contiene el modelo dinámico, el cual es necesario para realizar la simulación en Simulink [19]. La ley de control esta dada por:

$$\Gamma = K_{pi}(q_{des} - q_{real}) + K_{vi}(q_{des} - q_{real}) \quad (4.2)$$

Donde $K_p = K_c$, $K_i = 1/T_i$ y $K_d = T_d$ siendo K_c , T_i y T_d la ganancia proporcional, el tiempo integral del controlador y el tiempo derivativo del controlador respectivamente.

Debido a lo mencionado anteriormente la sintonización de este control se realizó para 3GDL

4.2.6. Simulación en MATLAB - Simulink

Para realizar la simulación del robot en un ambiente como Matlab/Simulink se necesitan valores de los parámetros dinámicos del mecanismo. De la misma manera que se usaron los bloques para el control CTC, se usaron en este caso, de la siguiente manera:

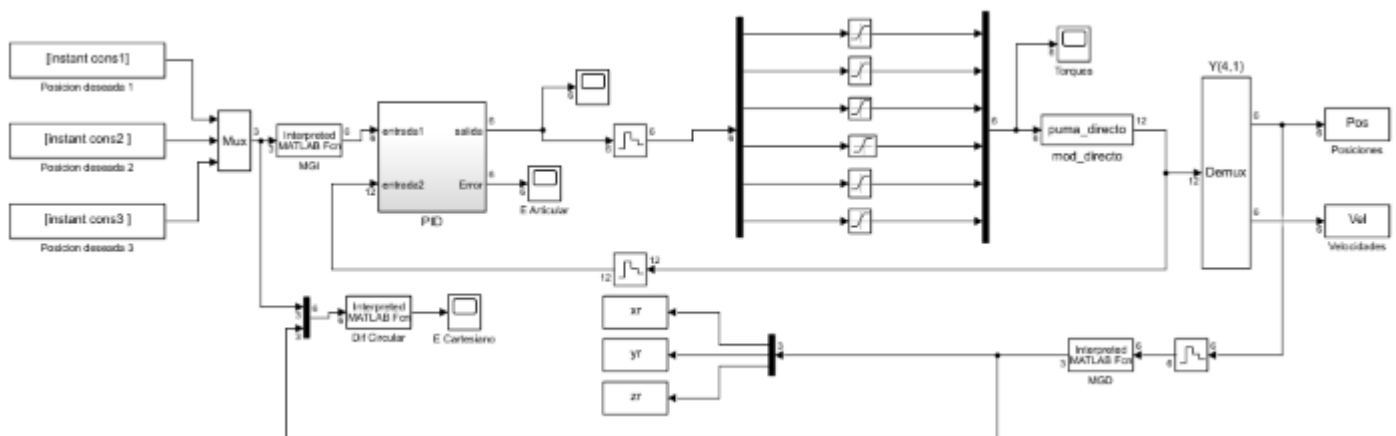


Figura 4.12: Esquema control PID cartesiano.

Fuente: Modificada de [19]

En el subsistema encontramos el control PD, la cual su sintonización fue a prueba y error para 3GDL ya que para controlar 6GDL por este modo es muy dificultoso y se necesita de otros métodos de control .

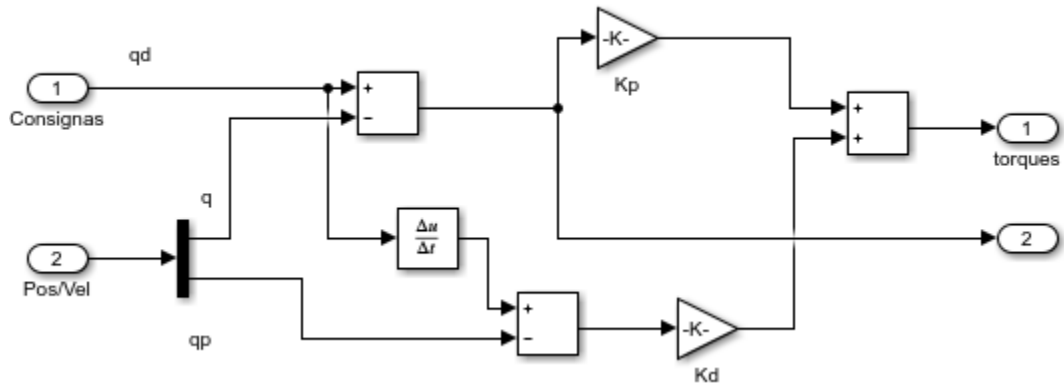


Figura 4.13: Control PID para robot Puma.
Fuente: Modificada de [19]

Se obtiene un error cartesiano lo mas pequeño posible mostrado a continuación:

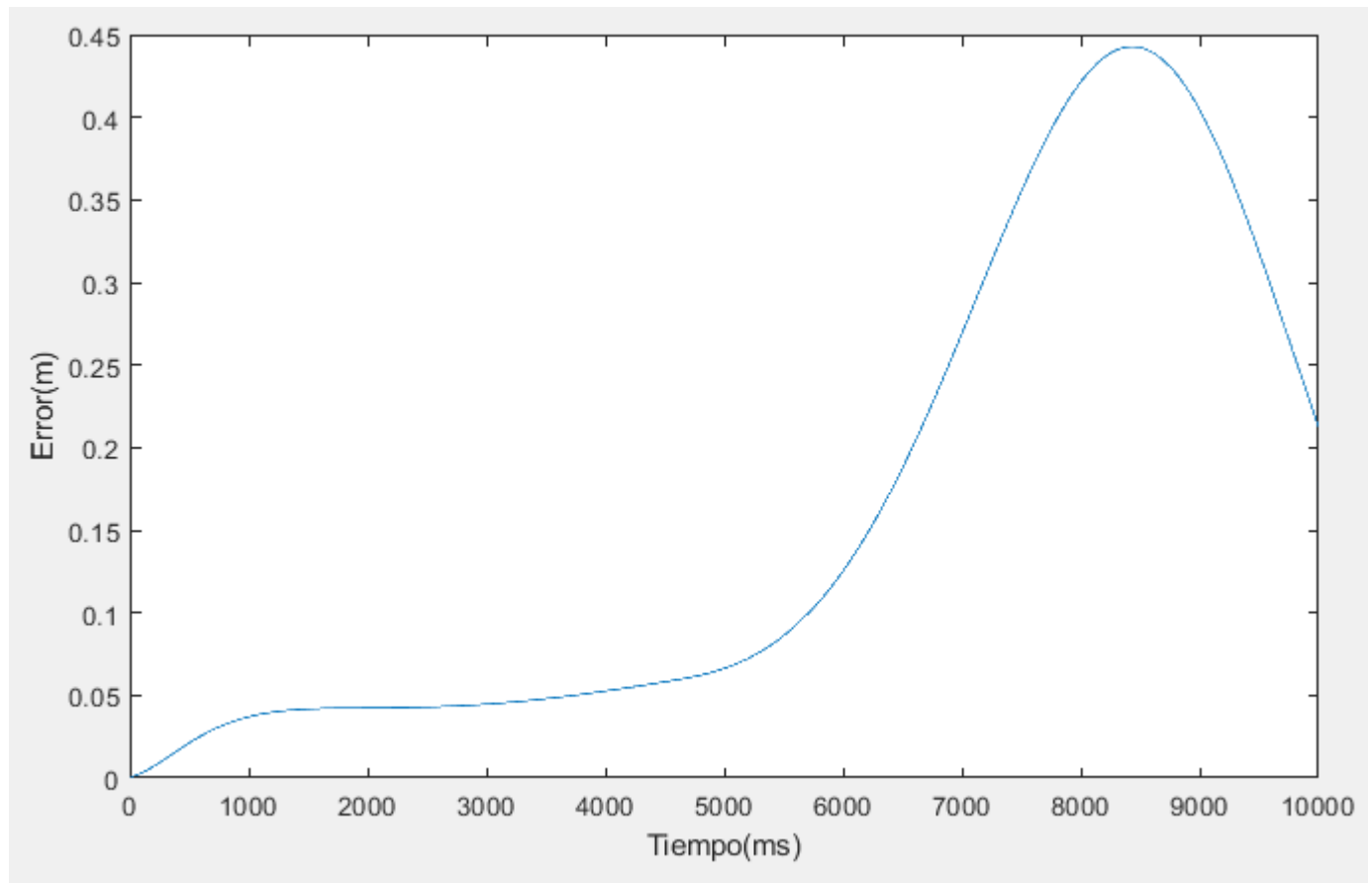


Figura 4.14: Error cartesiano PID.

Fuente: Diseño propio.

Seguidamente se obtiene la consigna deseada, donde se aprecia que el robot intenta seguir la trayectoria que es la consigna círculo, pero se analiza que no es demasiado eficiente por su complejidad de sintonización con este método; esto se muestra a continuación:

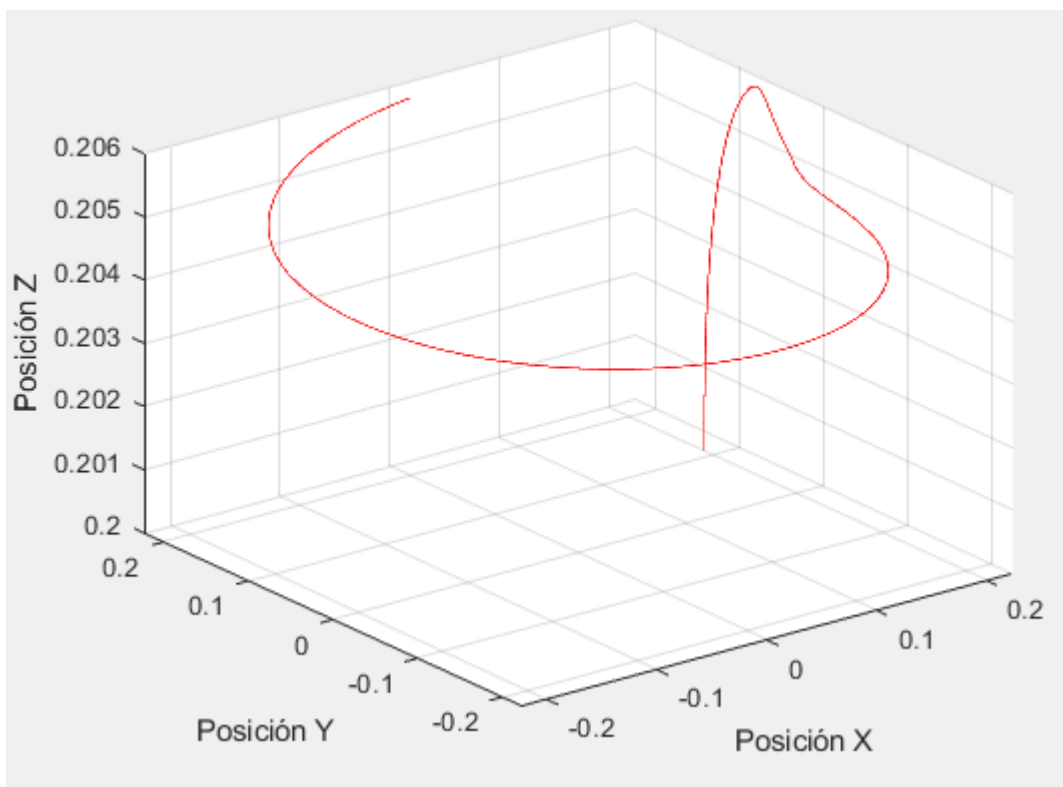


Figura 4.15: Comportamiento de control PID cartesiano frente a una trayectoria circular.
Fuente: Diseño propio.

Para el sistema PID articular se realizó la sintonización con 3GDL del robot ya que para este tipo de controles tiene una mayor dificultad en el momento de realizar la sintonización por el método de prueba y error.

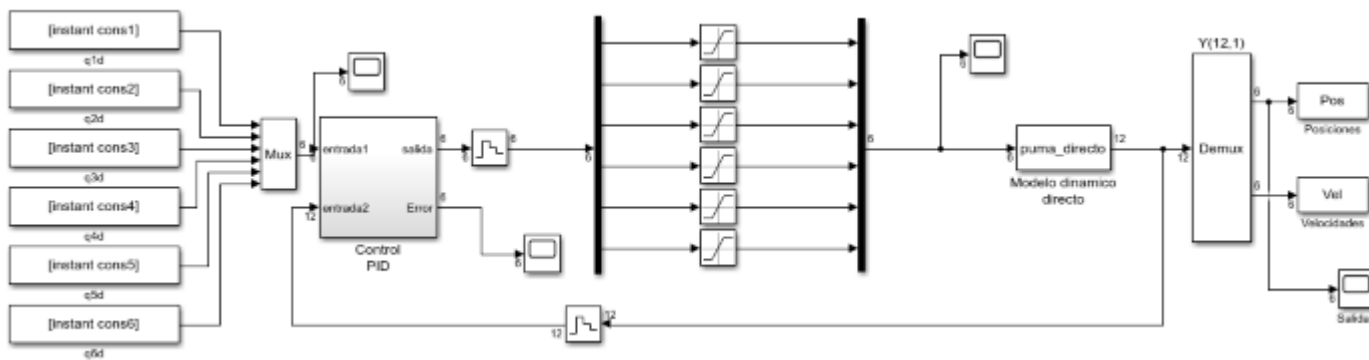


Figura 4.16: Esquema PID articular robot Puma.
Fuente: Diseño propio.

De la misma manera que en los anteriores sistemas se tiene el control para el PID articular.

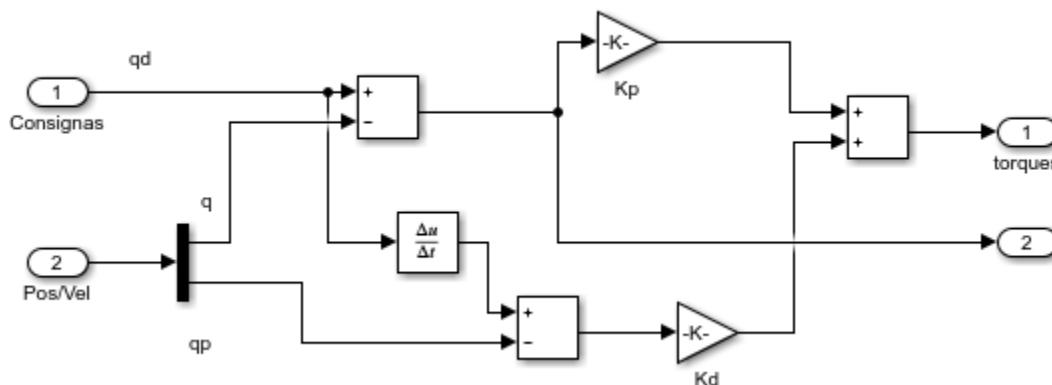


Figura 4.17: Control PID articular robot Puma.
Fuente: Tomada de [19].

Como se observa en la figura 4.2.6 se realizó la sintonización para las primeras 3 articulaciones teniendo como resultado el menor error posible.

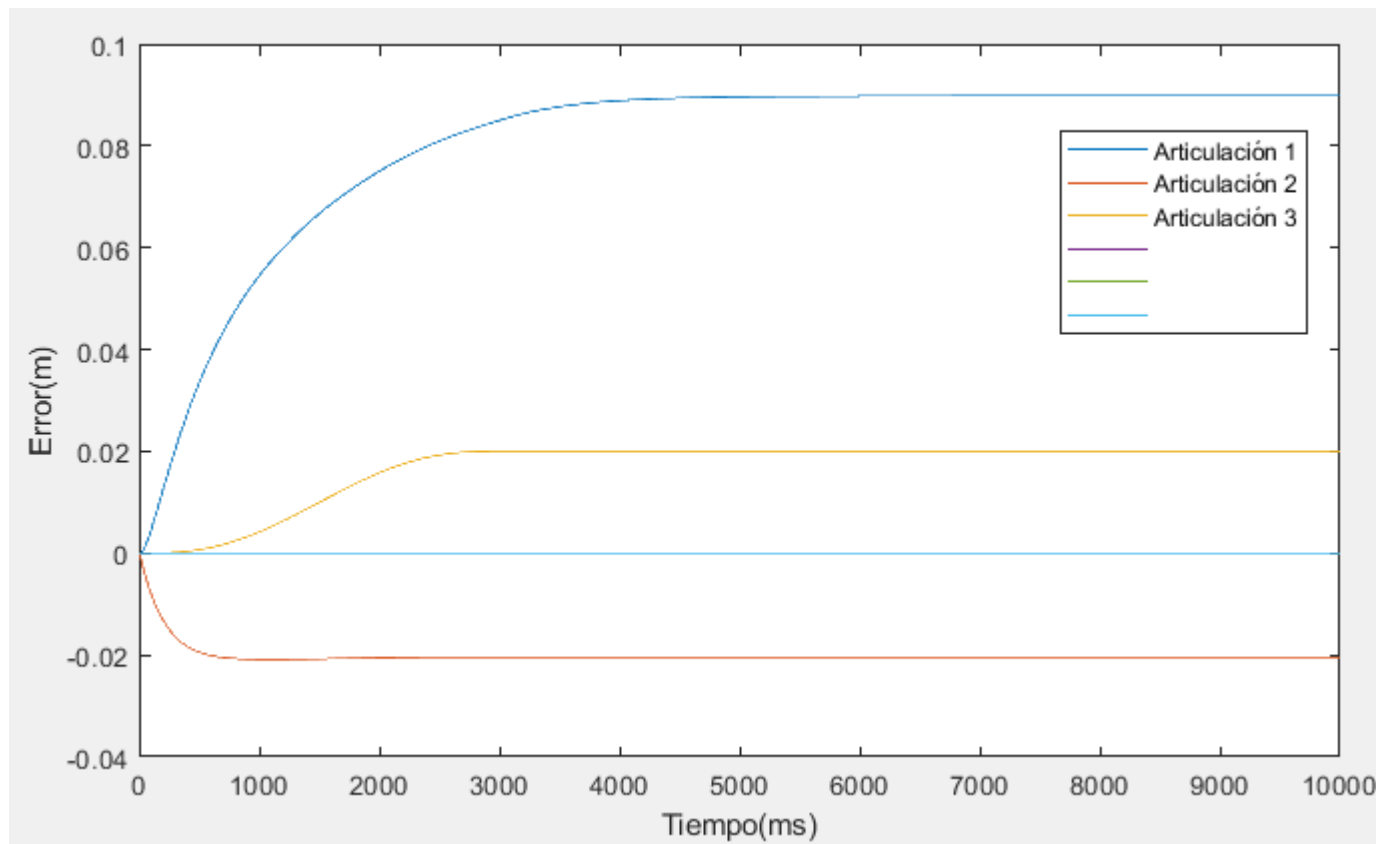


Figura 4.18: Error articular PID articular.
Fuente: Diseño propio.

Capítulo 5

Pruebas y resultados

5.1. Resultados de implementación

Se da por finalizado el proceso de desarrollo del prototipo de robot Puma de seis grados de libertad sobre el que se realizaría la implementación y prueba de las estrategias de control. Como resultado del proceso de ensamblaje, se presenta en la figura 5.1. la evidencia fotográfica del ensamblaje físico realizado.



Figura 5.1: Prototipo robot Puma terminado.
Fuente: Diseño propio.

Como paso inicial para realizar las pruebas de seguimiento de trayectorias y evaluar comparativamente el desempeño de los controladores es necesario definir y caracterizar las pruebas que se realizarían; el marco de referencia para mediciones y las consignas a ser realizadas.

5.1.1. Marco de referencia

Se considera un marco de referencia común el punto en que todos los actuadores rotacionales se encuentran en una posición angular de cero grados, es decir dejando todas las partes del robot y de las cadenas cinemáticas con una disposición perpendicular a la base del robot.

5.1.2. Trayectorias establecidas

En relación a las trayectorias deseadas se plantea como requerimiento el seguimiento de la consigna de grado cinco para el plano articular, circular para el plano cartesiano y por ultimo el análisis de la consigna *pick and place*.

5.1.3. Trayectoria grado cinco

Es una consigna de quinto grado que se define para cada una de las articulaciones del robot, sean rotoides o prismáticas que para este caso todas sus articulaciones son rotoides.

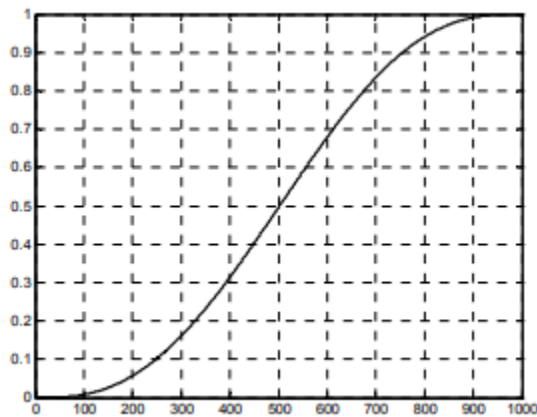


Figura 5.2: Trayectoria grado cinco.
Fuente: Tomada de [19]

5.1.4. Trayectoria Circular

Es una consigna que se define sobre dos ejes, normalmente x e y. Permite dibujar un círculo de radio y centro definidos por el usuario. Se debe tener en cuenta el espacio de trabajo del robot, pues fuera de este espacio el robot no podrá seguir la trayectoria [19].

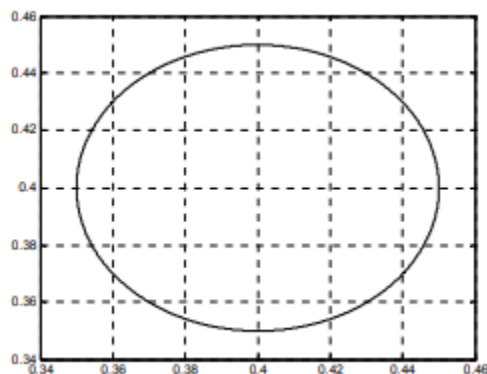


Figura 5.3: Trayectoria Circular.
Fuente: Tomada de [19].

5.1.5. Trayectoria pick and place

Como trayectoria final se emulan las necesidades de una trayectoria de tipo *pick and place*. Inicialmente la parte terminal se desplaza positivamente en el eje Z emulando el comportamiento tras haber recogido un elemento, posteriormente el robot desplaza este elemento a través del plano XY manteniéndose hasta alcanzar la ubicación deseada y seguidamente el robot desciende sobre el eje Z para posicionar en su ubicación final. La trayectoria deseada se muestra a continuación.

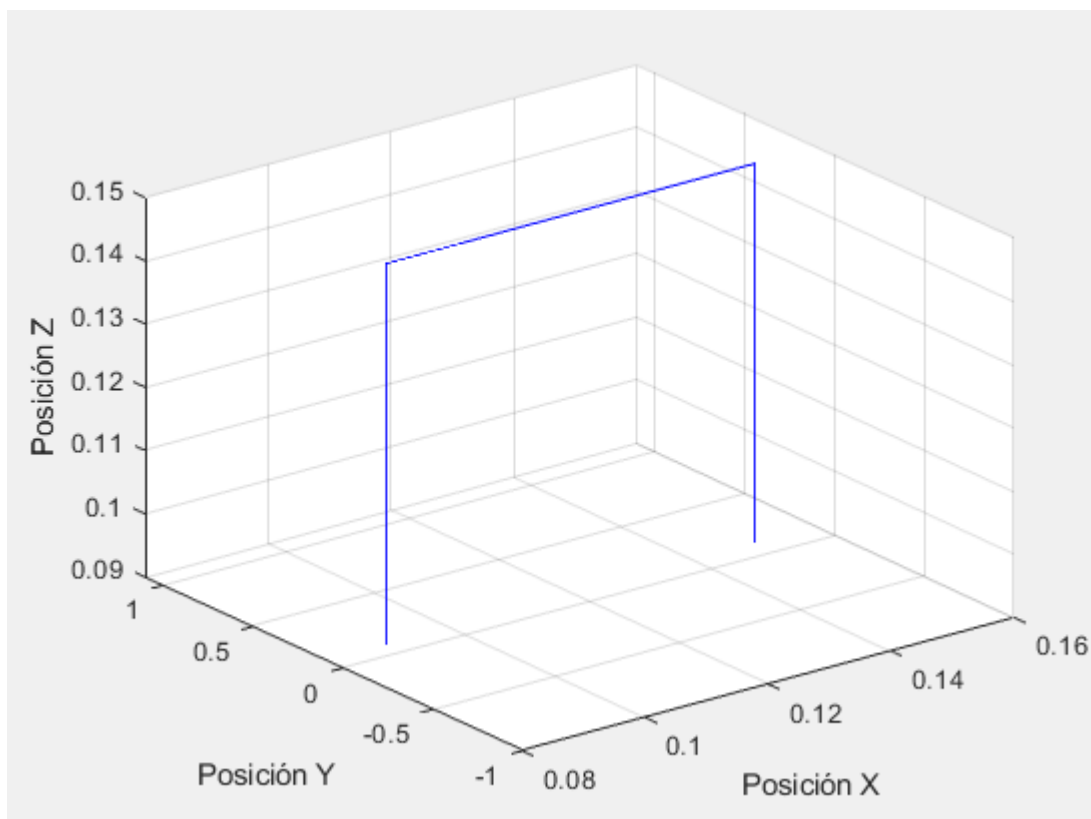


Figura 5.4: Trayectoria pick and place.
Fuente: Diseño propio.

5.2. Desempeño de control CTC y control PID

En esta sección se plasma los resultados obtenidos de la implementación del control CTC y PID en el prototipo de robot Puma frente al seguimiento de diferentes trayectorias, ilustrando el comportamiento en términos de posición articular, error cartesiano y comportamiento en términos de velocidad. Es importante señalar que debido a la ausencia de sensores que permitan realizar mediciones de velocidad en el robot, estos valores se obtuvieron a partir de los datos de lectura de posición articular en el inicio de las cadenas cinemáticas. Cabe aclarar que es posible encontrar diferencias introducidas por el modelo geométrico directo todo esto causado por que el cálculo se hizo por métodos numéricos.

5.3. Desempeño de control CTC

5.3.1. Comportamiento robot real trayectoria grado cinco CTC

A continuación primero se detalla como se hace el seguimiento de la consigna grado cinco del robot real, seguido de una secuencia de imágenes donde se observa como se comporta el robot.

Se observa que en la imagen 5.5 el error que se obtuvo al realizar el seguimiento de la trayectoria, como se observa la gráfica los valores de error son óptimos para su funcionamiento.

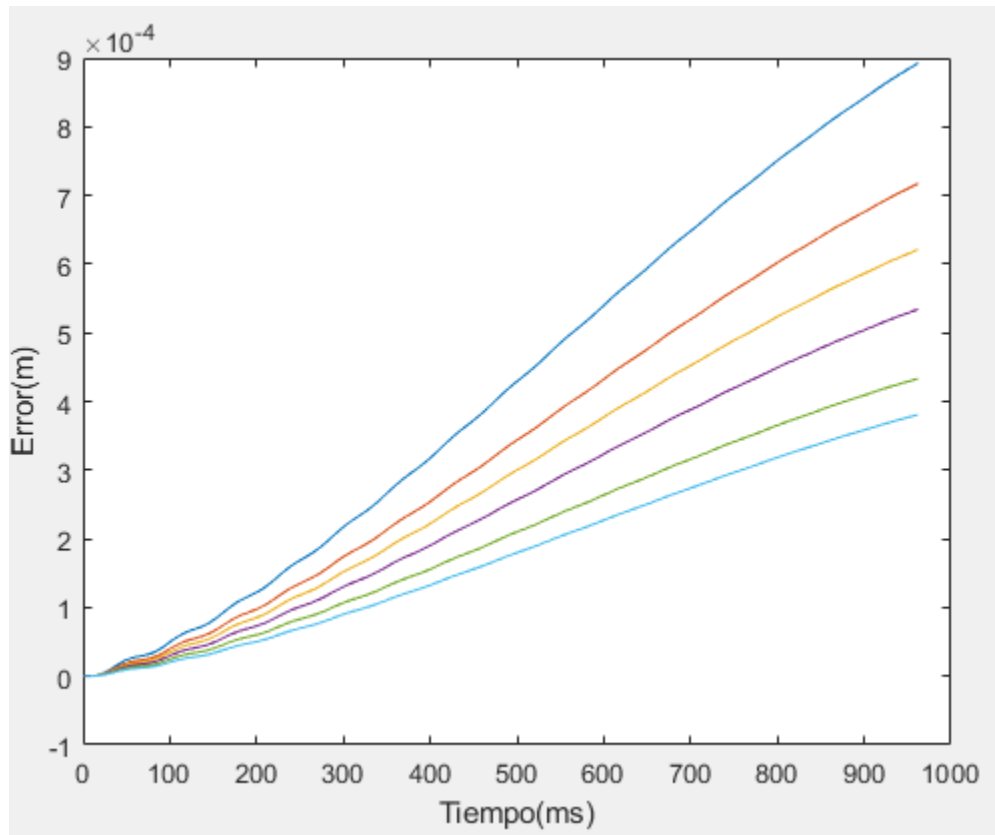


Figura 5.5: Error articular CTC con trayectoria grado cinco.
Fuente: Diseño propio.

En la siguiente secuencia de imágenes se muestra la trayectoria grado cinco con el robot Puma.

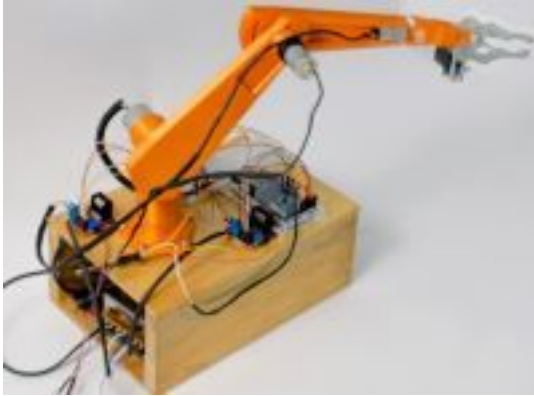


Figura 5.6: Trayectoria lineal.
Fuente: Diseño propio.

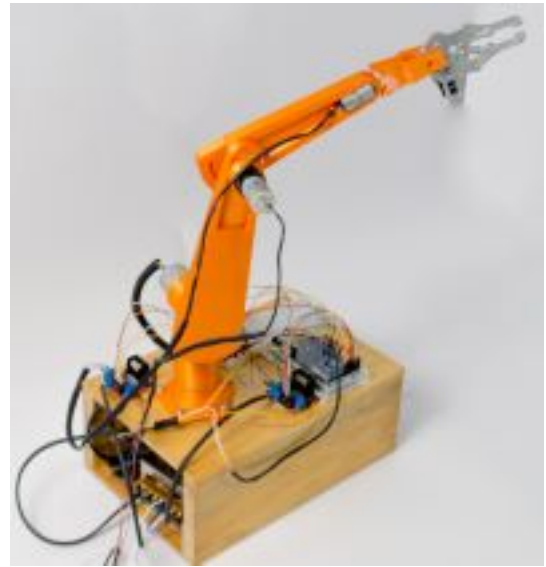


Figura 5.7: Trayectoria lineal.
Fuente: Diseño propio.

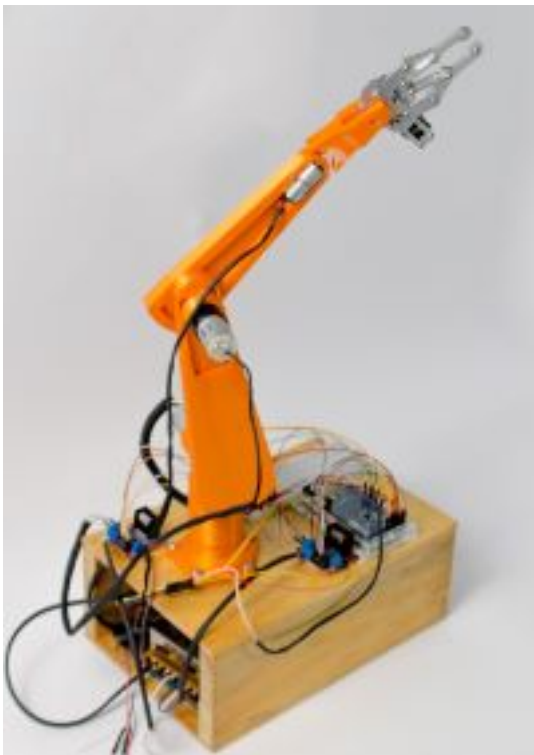


Figura 5.8: Trayectoria lineal.
Fuente: Diseño propio.



Figura 5.9: Trayectoria lineal.
Fuente: Diseño propio.

5.3.2. Comportamiento robot real trayectoria circular CTC

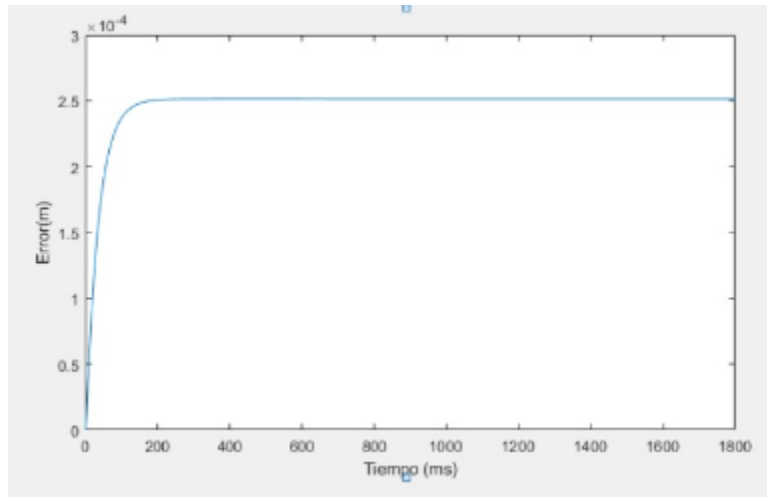


Figura 5.10: Error CTC Cartesiano del robot real con trayectoria circular.
Fuente: Diseño propio.

5.4. Desempeño control PID

5.4.1. Comportamiento robot real trayectoria grado cinco PID

Esta gráfica se obtuvo enviando la trayectoria grado cinco por medio de Matlab-Simulink al arduino realizando su control para las 3 primeras articulaciones ya que siendo un robot educativo se complicaba la sintonización de este para las 6 articulaciones.

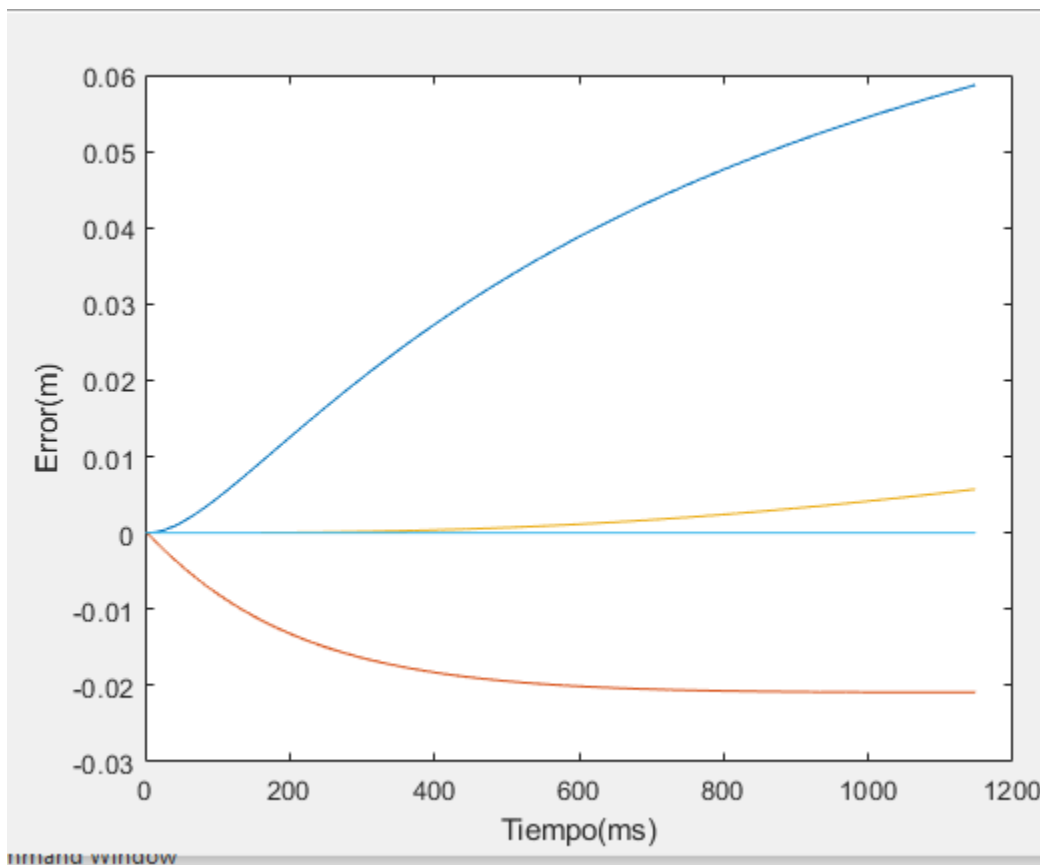


Figura 5.11: Error PID articular del robot real con trayectoria grado cinco.

Fuente: Diseño propio.

5.4.2. Comportamiento robot real trayectoria circular PID

Se evalúa la respuesta del sistema ante la trayectoria circular buscando el mayor grado de correspondencia entre consigna y valor obtenido para evaluar su rendimiento bajo criterios cuantificables y visuales.

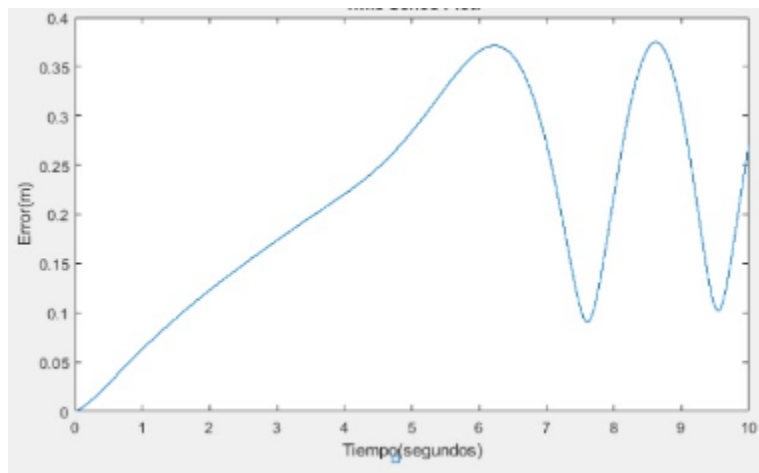


Figura 5.12: Error cartesiano PIS del robot real con trayectoria circular.
Fuente: Diseño propio.

En la anterior gráfica se puede observar la señal del error cartesiano en la cual tiene la sintonización para tres articulaciones como se menciono anteriormente ya que el control PID de mas de tres grados de libertad requiere otros métodos para lograr sintonizar de manera óptima las demás articulaciones.

Como se puede evidencia, el error cartesiano del modelo presenta un buen comportamiento pero dado a que no ha sido sintonizada sus demás articulaciones este tiende a descontrolarse.

5.5. Desempeño de control CTC en trayectoria pick and place

Se evalúa la respuesta del sistema ante la tercera trayectoria buscando el mayor grado de correspondencia entre consigna y valor obtenido para evaluar su rendimiento bajo criterios visuales y cuantificados.

5.5.1. Trayectoria pick and place con control CTC Matlab-Simulink

En esta parte se muestra la trayectoria realizada por el robot en rojo y la trayectoria deseada en azul; una mayor precisión en los dos segmentos iniciales y una perdida en el descenso final. Se recalca que las estimaciones de posición cartesiana a través del modelo dinámico no son las mas precisas.

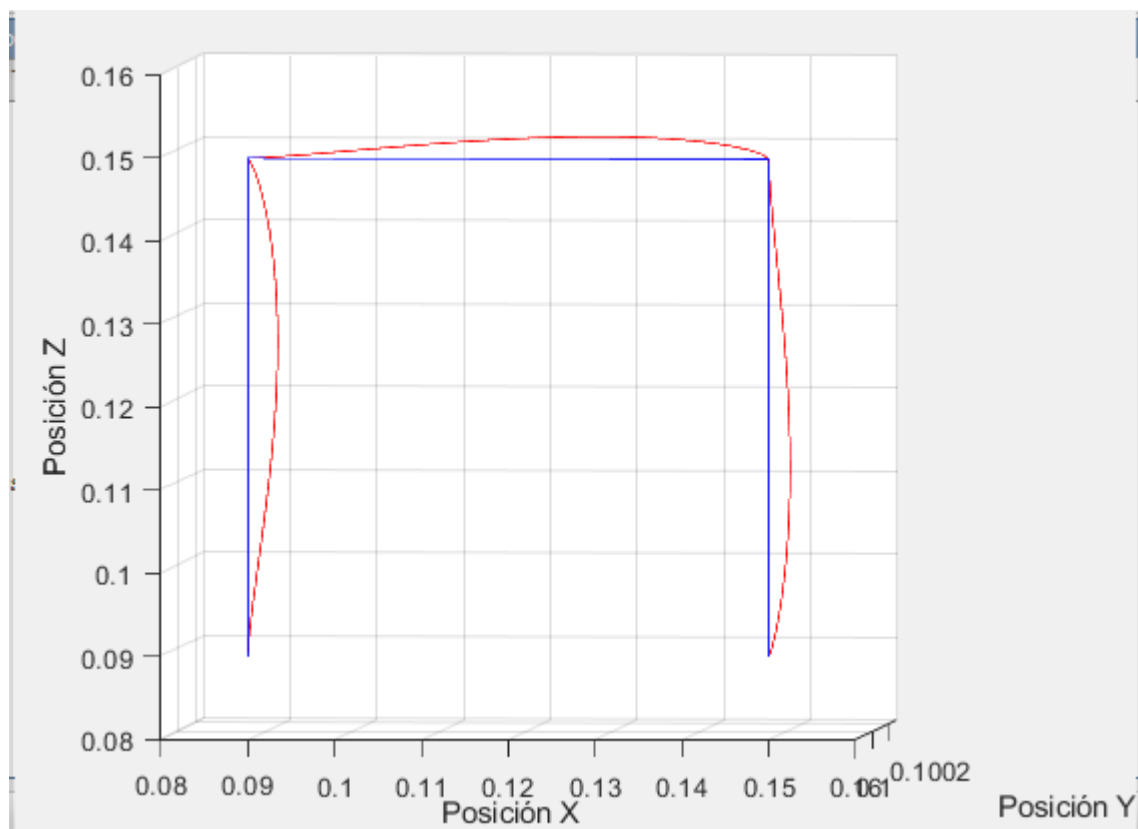


Figura 5.13: Trayectoria pick and place.
Fuente: Diseño propio.

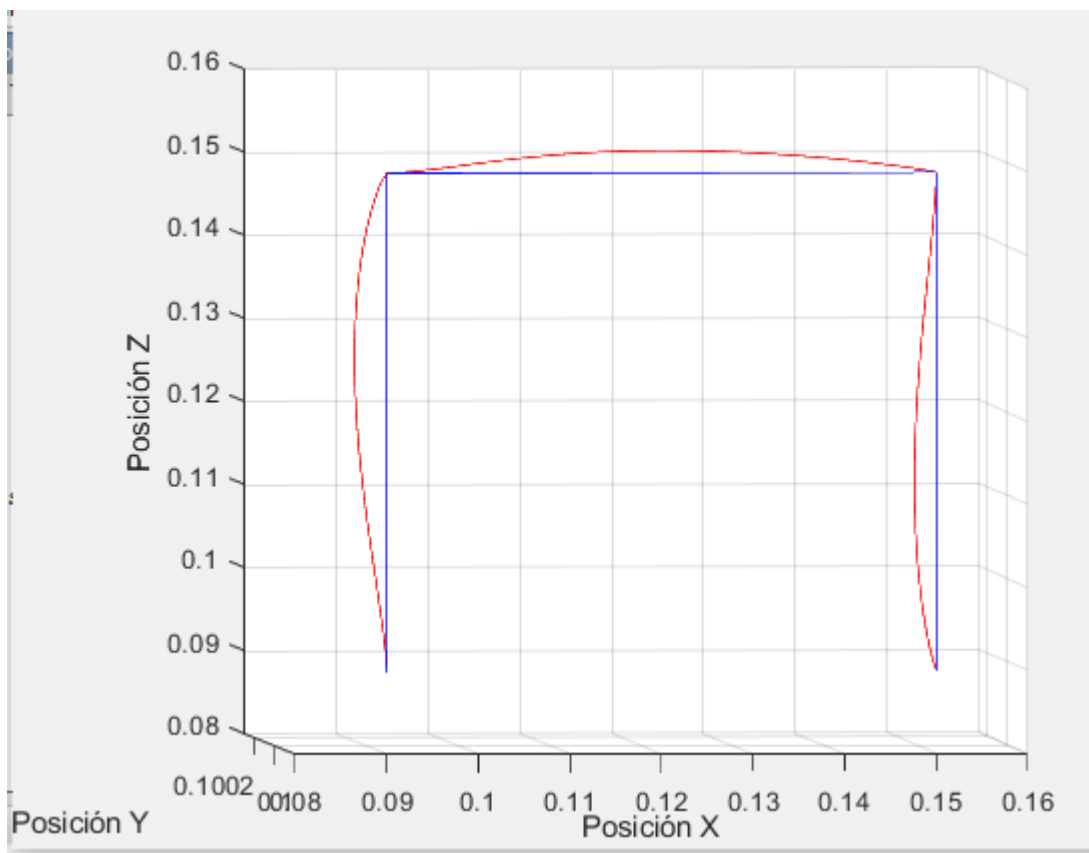


Figura 5.14: Trayectoria pick and place.
Fuente: Diseño propio.

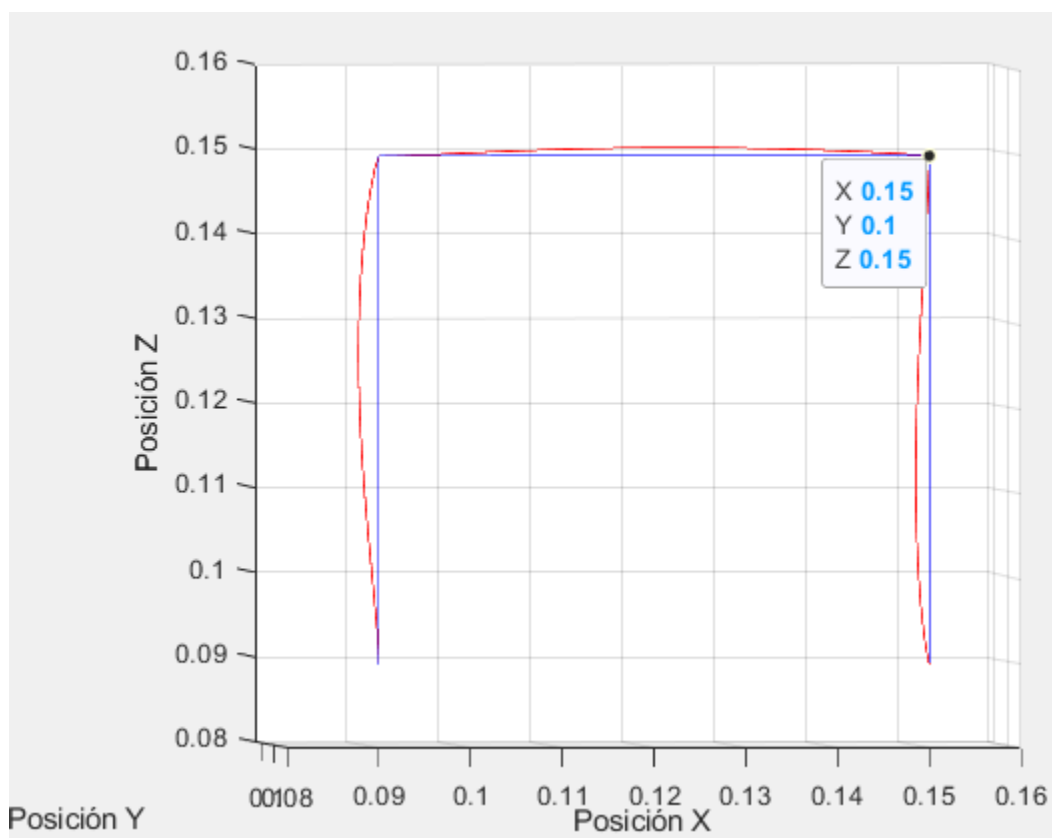


Figura 5.15: Trayectoria pick and place.
Fuente: Diseño propio.

A diferencia del comportamiento en otras consignas como la lineal en el movimiento articular del robot, el seguimiento de la consigna *pick and place* no presenta grandes inconvenientes debido a los pocos cambios de dirección articular presente en esta trayectoria.

A continuación se presenta el error cartesiano

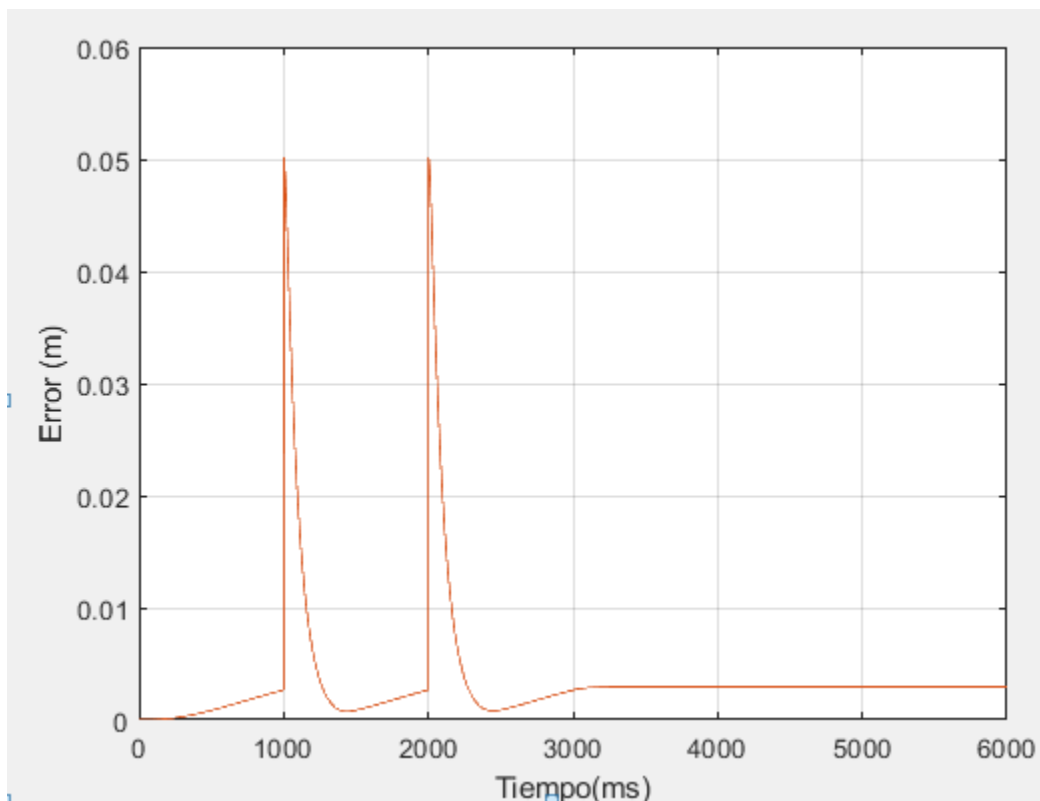


Figura 5.16: Error CTC cartesiano pick and place.
Fuente: Diseño propio.

5.6. Desempeño de control PID en trayectoria pick and place

Para realizar la simulación del control PID en el seguimiento de la trayectoria *pick and place* se muestra a continuación una gráfica de error ya que al momento que el robot intenta realizar la trayectoria deseada no encuentra los puntos ingresados y comienza a buscar puntos por todo el espacio de trabajo y como resultado se obtiene la siguiente grafica

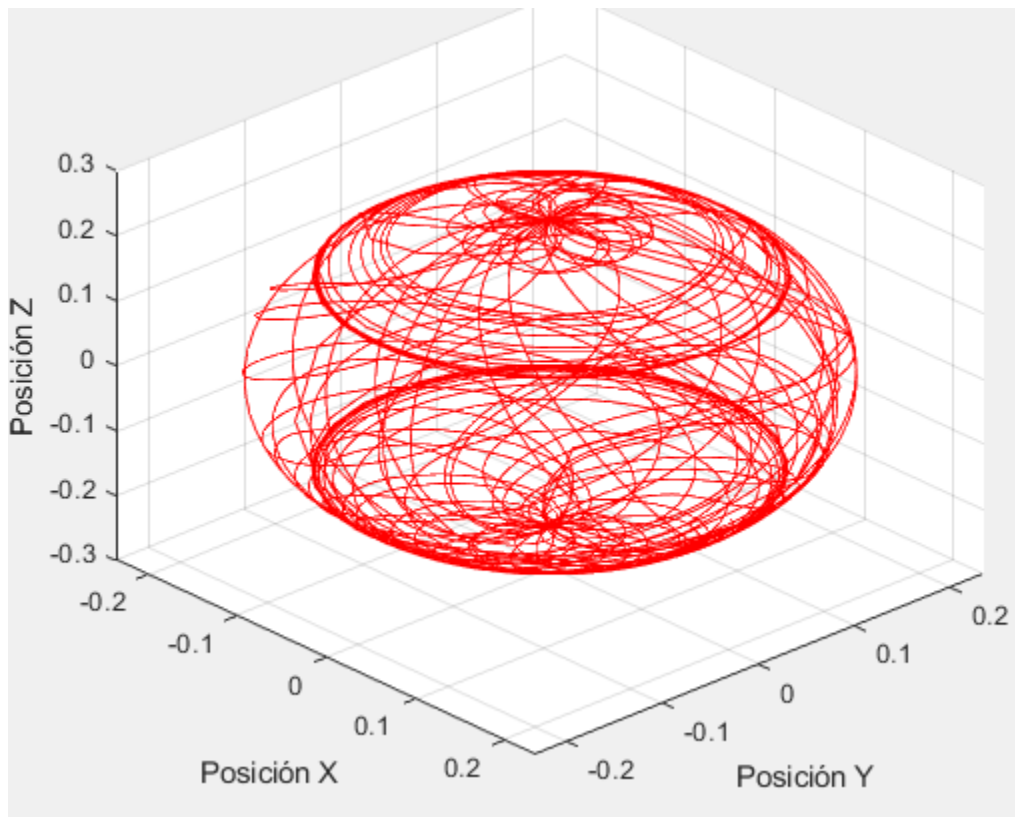


Figura 5.17: Trayectoria pick and place PID en Matlab-Simulink.
Fuente: Diseño propio.

5.7. Análisis de velocidad

Al igual que en las anteriores trayectorias es difícil hacer un análisis y comparación de aceleraciones en el control CTC aplicada al prototipo, esto se debe al alto nivel de ruido de la señal, sin embargo se observó que el control CTC tienen un comportamiento aceptable en cuanto al seguimiento de la trayectoria.

Capítulo 6

Resultados

6.1. Conclusiones

Durante el desarrollo de este proyecto se modela el robot Puma de seis grados de libertad basados en una arquitectura ya diseñada, para su desarrollo nos apoyamos en el software SolidWorks, donde se modela cada uno de los eslabones que fueron sacados en una impresora 3D, seguidamente se sacaron los parámetros dinámicos del robots y se implementan los dos métodos de control en simulación: PID (Proporcional, Derivativo e Integral) y CTC (Control por Torque Calculado) en ambas instancias se obtienen las partes articulares y cartesianas para poder ser simulado.

El control de robot Puma de seis grados de libertad se logró satisfactoriamente gracias al estudio de las diferentes áreas de la mecatrónica; como son: la electrónica, la mecánica y control, se concluyó en cada una de las áreas mencionadas su eficiencia o ineficiencia para el control con seis grados de libertad.

Se observó satisfactoriamente que el controlador articular muestra el movimiento para cada articulación y el cartesiano muestra el movimiento del robot con diferentes trayectorias por ejemplo: trayectorias circulares, lineales, en tres dimensiones, etc. Se hacen diferentes pruebas de movimiento con consignas deseadas por medio del software Matlab/Simulink estas consignas dependen si para el control CTC o para el control PID obteniendo errores demasiado pequeños.

Se concluye que las dos estrategias de control comparten una misma finalidad, su diferencia es en cuanto la sintonización de estos dos controladores dado que el PID es más compleja su implementación y dado a esto se realizó para 3GDL como se menciona en el desarrollo del proyecto; para su buen desempeño y una buena sintonización de los 6 GDL del robot tipo Puma se utilizan diferentes métodos al que es usado en este trabajo. En contraste del controlador CTC que es muy manejable a la hora de ser sintonizado, simulado e implementado, también su funcionamiento fue ideal en las pruebas realizadas siguiendo las consignas deseadas de una manera ideal y su implementación fue importante para observar que tan eficiente son estos tipos de controladores.

6.2. Trabajos futuros

Se contextualiza el presente trabajo de grado dentro del marco de la implementación de controladores en robots, se identifican varios puntos que a futuro pueden aportar continuidad en esta línea

de investigación. Dentro de los trabajos futuros propuestos se encuentran:

- Realizar la sintonización del PID con los seis grados de libertad.
- Mejorar la precisión del robot ajustando todo tipo de juegos mecánicos que podría tener.
- Se puede conectar sensores de otro tipo como son de visión, ultrasónicos y de fuerza, para experimentar con otro tipo de técnicas de control en la estructura del robot y darle mayores capacidades de interacción con el medio al brazo robótico.
- Realizar una interfaz para interacción con el robot en un software más avanzado como por ejemplo Unity, para lograr así obtener un mejor desempeño con el robot siendo este un sistema independiente.
- Se puede utilizar otro tipo de protocolos de comunicación entre el hardware y realizar aplicaciones con las librerías matplotlib, NumPy y PyQt para llevar estadísticas de los movimientos y trazados de las trayectorias de forma visual.

Bibliografía

- [1] Yun Li, Kiam Heong Ang y Gregory CY Chong. “Patents, software, and hardware for PID control: an overview and analysis of the current art”. En: *IEEE Control Systems Magazine* 26.1 (2006), págs. 42-54.
- [2] Isidro Zabalza y Javier Ros. “Aplicaciones actuales de los robots paralelos”. En: *8th Latin American Congress of Mechanical Engineering. Cusco, Perú*. Vol. 3. 9. 2007.
- [3] Cristina Urdiales, Antonio Bandera y Francisco Sandoval. “Historia y tendencias actuales de la robótica”. En: (1999).
- [4] Leonardo Enrique Solaque Guzmán, Manuel Alejandro Molina Villa y Edgar Leonardo Rodríguez Vásquez. “Seguimiento de trayectorias con un robot móvil de configuración diferencial”. En: *Ingenierías USBMed* 5.1 (2014), págs. 26-34.
- [5] Pablo Torró Lluch. “Robot articulado de 5 GDL didáctico diseñado para fabricarse con impresora 3D.” Tesis doct. 2015.
- [6] Andrés Felipe Araque Mora, Francisco Javier Pedroza Fernández y col. “Desarrollo de un robot manipulador de 6 grados de libertad con tecnologías abiertas”. B.S. thesis. Universidad Autónoma de Occidente, 2019.
- [7] Héctor A Moreno, Roque Saltaren, Isela Carrera, Lisandro Puglisi y Rafael Aracil. “Índices de desempeño de robots manipuladores: una revisión del estado del arte”. En: *Revista Iberoamericana de Automática e Informática industrial* 9.2 (2012), págs. 111-122.
- [8] Germán Darío Buitrago Salazar y Olga Lucía Ramos Sandoval. “Sistema teledirigido de un brazo robótico de 4 grados de libertad aplicando visión de máquina”. En: *Revista EIA* 12.24 (2015), págs. 121-219.
- [9] S. Bennett. “Development of the PID Controller, Control Systems, IEEE, vol. 13,” en: (1994).
- [10] K. D. Barón y C. B. Pinilla. “Generalidades de robots paralelos,” *Visión electrónica*. En: (2016).
- [11] Ramon Garcia Cortes. “Análisis y desarrollo de un controlador de movimiento para un robot PUMA basado en FPGA”. En: (2014).
- [12] Antonio Barrientos, Luis Felipe Peñin, Carlos Balaguer y Rafael Aracil. *Fundamentos de robótica*. Vol. 2. McGraw-Hill Madrid, 2007.
- [13] S Torres y JA Méndez. “Seguimiento de trayectorias en robots manipuladores: revisión de soluciones y nuevas propuestas”. En: *Revista Iberoamericana de Automática e Informática Industrial RIAI* 6.4 (2009), págs. 80-92.
- [14] Nouredine BALI. “Controles predictivos aplicados al robot PUMA y rodamiento magnético activo”. En: (2007).

- [15] E Freund. “Fast nonlinear control with arbitrary pole-placement for industrial robots and manipulators”. En: *The International Journal of Robotics Research* 1.1 (1982), págs. 65-78.
- [16] Mark W Spong y Mathukumalli Vidyasagar. *Robot dynamics and control*. John Wiley & Sons, 2008.
- [17] Bruno Siciliano y Oussama Khatib. *Springer handbook of robotics*. Springer, 2016.
- [18] Duy Nguyen-Tuong, Matthias Seeger y Jan Peters. “Computed torque control with nonparametric regression models”. En: *2008 American Control Conference*. IEEE. 2008, págs. 212-217.
- [19] A Vivas y V Mosquera. “Predictive functional control of a PUMA robot”. En: *Conference Proceedings*. 2005.
- [20] Rasika Yenorkar y UM Chaskar. “GUI based pick and place robotic arm for multipurpose industrial applications”. En: *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE. 2018, págs. 200-203.
- [21] Esmaeil Najafi y Mahdi Ansari. “Model-based design approach for an industry 4.0 case study: a pick and place robot”. En: *2019 23rd International Conference on Mechatronics Technology (ICMT)*. IEEE. 2019, págs. 1-6.
- [22] Mathieu Garon, Denis Laurendeau y Jean-François Lalonde. “A framework for evaluating 6-DOF object trackers”. En: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, págs. 582-597.
- [23] Varios autores. *PUMA (robot)*. [Web; accedido el 2-06-2022]. 2021. URL: [URL{https://es.wikipedia.org/wiki/PUMA_\(robot\)}](https://es.wikipedia.org/wiki/PUMA_(robot)).
- [24] Marina León, Ángel Marcos-Fernández y Juan Rodríguez-Hernández. “Impresión 3D con materiales elásticos”. En: (2019).
- [25] Arduino Pagina web. <https://store.arduino.cc/usa/>. 22 de Marzo de 2022.
- [26] Varios autores. *Qué es UART*. [Web; accedido el 26-05-2022]. 2022. URL: [URL{https://www.rohde-schwarz.com/es/productos/test-y-medida/osciloscopios/educational-content/que-es-uart_254524.html#:~:text=UART%20\(universal%20asynchronous%20receiver%20%2F%20transmitter,y%20recibir%20en%20ambas%20direcciones.}](https://www.rohde-schwarz.com/es/productos/test-y-medida/osciloscopios/educational-content/que-es-uart_254524.html#:~:text=UART%20(universal%20asynchronous%20receiver%20%2F%20transmitter,y%20recibir%20en%20ambas%20direcciones.})
- [27] José Miguel López Sánchez y col. “Control de posición y seguimiento de trayectorias para un robot neumático de estructura paralela”. En: (2012).
- [28] André Francisco Gonzáles Chávez. “Diseño del sistema de control de un Robot Tipo PUMA utilizando LÓGICA DIFUSA”. En: (2013).
- [29] SALVADOR ORTIZ SANTOS y col. “Sintonización de un controlador PID basado en un algoritmo heurístico para el control de un Ball and Beam”. En: (2018).
- [30] Nick J Killingsworth y Miroslav Krstic. “PID tuning using extremum seeking: online, model-free performance optimization”. En: *IEEE control systems magazine* 26.1 (2006), págs. 70-79.
- [31] Luis Daniel Villanes Bedoya. “Diseño de un algoritmo PID sintonizado mediante lógica fuzzy para controlar un brazo robótico”. En: (2016).