



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Planeación de movimientos
para manipulación de
objetos en robots de servicio
doméstico**

TESIS

Que para obtener el título de

Ingeniera Eléctrica Electrónica

P R E S E N T A

Biddy Itzel González Jiménez

DIRECTOR DE TESIS

Dr. Marco Antonio Negrete Villanueva



Ciudad Universitaria, Cd. Mx., 2024

*Dedicado a Rodrigo Savage,
quien me mostró ese mundo
de robots que solo conocía en
las novelas de Issac Asimov
y que me inspiró a seguir el
camino de la ingeniería.
Gracias amigo.*

Agradecimientos

- A mi madre, pues sin ella no lo hubiera logrado. Gracias por todo tu apoyo, tu amor y paciencia conmigo.
- A Dalia, por motivarme a seguir mis metas aun cuando tenía ganas de renunciar.
- A mis hermanos Yotecatl y Tlacaélel porque con su compañía y sus consejos, los momentos difíciles se volvían mas amenos.
- A mi padre porque con los libros y videojuegos que me mostraba desde temprana edad despertó en mi la curiosidad y fascinación por la tecnología.
- Gracias al Dr. Savage por promover la robótica en el país y por invitarme a trabajar en el laboratorio de Bio-Robótica de la UNAM.
- Este trabajo se elaboró con apoyo del proyecto PAPIIT 102424 "Modelos lógico probabilísticos para el desarrollo de robots móviles autónomos".

Índice general

1. Introducción	6
1.1. Motivación	7
1.2. Planteamiento del problema	8
1.3. Hipótesis	8
1.4. Objetivos	9
1.4.1. Objetivo general	9
1.4.2. Objetivos particulares	9
1.5. Descripción del documento	9
2. Marco Teórico	11
2.1. Robots de servicio	11
2.1.1. Clasificación	11
2.1.2. El manipulador robótico	13
2.2. Planeación para manipulación	13
2.3. Trabajos relacionados	15
2.3.1. <i>MoveIt!</i>	18
2.4. Espacio de configuración	18
2.4.1. Articuciones robóticas	18
2.4.2. Espacio de tareas y espacio de trabajo	19
2.5. Representación de posición y orientación	19
2.5.1. Posición y desplazamiento	19
2.5.2. Orientación y rotación	20
2.5.3. Composición de rotaciones	21
2.5.4. Parametrización de rotaciones	21
2.5.5. Transformaciones homogéneas	22
2.5.6. Árbol de transformaciones homogéneas	23
3. Detección y localización de objetos	26
3.1. Adquisición de datos con <i>Kinect</i>	27
3.2. Segmentación de imagen	29

3.2.1.	Segmentación por agrupamiento de píxeles en base al color	29
3.2.2.	Agrupamiento por color usando el algoritmo <i>k-Medias</i>	29
3.3.	Detección de objetos en la escena	30
3.3.1.	Pre-procesamiento de la imagen	31
3.3.2.	Cuantización por color de la imagen	31
3.3.3.	Detección de contornos	32
3.4.	Orientación mediante PCA	34
3.4.1.	Estimación de la posición del objeto	34
3.4.2.	Estimación de la orientación del objeto	35
3.4.3.	Procesamiento de la nube de puntos	37
3.4.4.	Construcción del sistema coordenado del objeto	38
3.4.5.	Estimación de la geometría del objeto	40
4.	Planeación de movimientos	43
4.1.	Descripción del manipulador	44
4.1.1.	Puntos de singularidad	44
4.2.	Cinemática del manipulador	45
4.2.1.	Cinemática Directa	45
4.3.	Cinemática inversa	46
4.4.	Cinemática diferencial	47
4.5.	Matriz Jacobiana	47
4.5.1.	Jacobiano Geométrico	47
4.5.2.	Jacobiano Analítico	48
4.6.	Algoritmo de cinemática inversa	49
4.6.1.	Matriz Jacobiana inversa	50
4.6.2.	Soluciones cerradas	51
4.6.3.	Soluciones numéricas	51
4.6.4.	El método de Newton-Raphson	51
4.6.5.	Cálculo numérico del Jacobiano	52
4.6.6.	Planeación de trayectorias para mejorar la convergen- cia de N-R	53
4.7.	Planeación de trayectorias	54
4.8.	Reglas de agarre propuestas	57
4.8.1.	Utilización de la caja delimitadora del objeto	57
4.8.2.	Utilización de la orientación del objeto	57
4.8.3.	Inclinación del objeto	57
4.8.4.	Utilización del sistema coordenado del objeto	58
4.8.5.	Tipos de agarre definidos	59
4.8.6.	Implementación de las reglas de agarre	60
4.8.7.	Regla de agarre para objetos tipo cilindro/barra	60

4.8.8. Objeto plano/caja	64
4.8.9. Objeto esférico/cuboide	67
4.9. Modelo Dinámico del Manipulador	68
4.9.1. Control proporcional derivativo con compensación de gravedad(PD+)	69
4.9.2. Implementación del controlador	70
5. Resultados	72
5.1. Implementación en ROS	72
5.1.1. El simulador Gazebo	72
5.1.2. Nodos de ROS creados	73
5.2. Pruebas del control de posición	74
5.3. Pruebas de agarre de objetos	74
5.4. Integración del sistema a las tareas de servicio	77
5.4.1. Pruebas para la liga Robocup@Home	85
6. Discusión	87
6.1. Conclusiones	87
6.2. Trabajo futuro	88

Capítulo 1

Introducción

En las últimas décadas los robots de servicio han extendido su presencia a múltiples áreas del sector de servicios. Esto se debe al rápido desarrollo de la tecnología en áreas de la electrónica y la computación, en consecuencia se tienen robots capaces de realizar tareas con un grado cada vez más alto de dificultad. La lista de aplicaciones de servicio crece debido a que algunas de las actividades humanas más peligrosas se encuentran en trabajos de servicio y porque estos trabajos son altamente robotizables y brindan mayores beneficios económicos a las empresas de servicios [10].

En el informe World Robotics 2021-Service Robots, presentado por la Federación Internacional de Robótica (IFR), se muestran las gráficas del total de ventas de robots de servicio en el año 2019 y 2020, estos datos reflejan el crecimiento a gran velocidad de la industria de los robots de servicio pues las ventas reportaron un incremento del 41 % en 2020 con respecto al año anterior.¹

Según este informe, entre las aplicaciones robóticas de servicio que experimentaron una mayor demanda se encuentran aquellas donde los robots brindan asistencia en hospitales, hoteles, restaurantes y tiendas de autoservicio, ya que debido a la reciente pandemia de Covid-19, los consumidores se volvieron más conscientes del riesgo de enfermedades infecciosas derivadas de las interacciones interpersonales y del contacto con productos alimenticios, además la necesidad de proporcionar servicios sin entrar en contacto con los consumidores acelera todavía más la adopción de estos robots.

Los robots de servicio que acompañan a las personas y les brindan asistencia son llamados robots de servicio doméstico. Estos robots tienen una interacción muy estrecha con los seres humanos y a diferencia de los robots industriales que se encuentran confinados en centros de trabajo con entornos

¹<https://ifr.org/ifr-press-releases/news/service-robots-hit-double-digit-growth-worldwide>

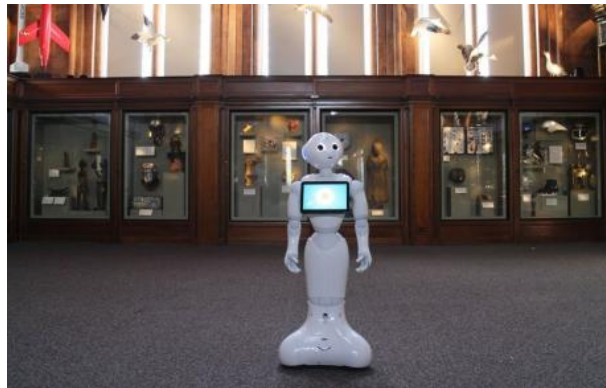


Figura 1.1: Robot de servicio usado en hotelería para brindar asistencia a los clientes (tomado de <https://www.aldebaran.com/es/industries/turismo>).

controlados, los robots de servicio doméstico deben contar con un conjunto de habilidades que les permitan desempeñar sus tareas en entornos arbitrarios y dinámicos. Dentro de ese conjunto se encuentra la habilidad de planeación de movimientos, necesaria para desempeñar tareas de manipulación de objetos.

La manipulación de objetos es una tarea simple para un ser humano, ya que puede tomar casi cualquier objeto, ya sea que lo haya visto o no. Pero para un robot involucra una gran dificultad, en la actualidad la manipulación de objetos se considera un problema de solución abierta. En este trabajo se propone un método con el cual un robot de servicio pueda realizar la planeación de movimientos y así desempeñar agarres exitosos de objetos desconocidos.

1.1. Motivación

En la actualidad los robots todavía encuentran dificultades al intentar manipular objetos cuyo tamaño, forma y material se desconoce, en consecuencia, el reconocimiento y la manipulación de objetos sigue siendo objeto de investigación.

En la literatura se pueden encontrar soluciones que utilizan aprendizaje de máquina y redes neuronales, tanto para determinar el mejor agarre así como para el reconocimiento de objetos, hay trabajos que efectúan el reconocimiento a partir de modelos precargados de objetos conocidos, este enfoque tiene el inconveniente de que puede llegar a consumir mucha memoria en la computadora del robot, además de que uno de los pasos en el reconocimiento del objeto suele ser la reconstrucción del modelo a partir de vistas desde diferentes ángulos, lo cual muchas veces no es posible por la cantidad

de tiempo que conlleva ese paso, o debido a las características del entorno o las restricciones en el hardware del robot.

También se ha abordado el problema utilizando algoritmos que involucran redes neuronales que necesitan de muchas horas de entrenamiento para determinar el mejor agarre de una poca cantidad de objetos, lo cual le resta capacidad de autonomía al robot al reducir su capacidad de acción si encuentra un objeto desconocido. Surge entonces la motivación de desarrollar un sistema de planificación de movimientos que dote a un robot de capacidad para determinar de forma autónoma, agarres rápidos y de bajo coste computacional.

1.2. Planteamiento del problema

El problema de manipulación robótica de objetos se enfoca en determinar la mejor orientación y posición del efector final para poder realizar la tarea de agarre de objetos, con el fin de trasladarlo de un lugar a otro. Este a su vez se divide en varios problemas: el problema de la cinemática directa, cinemática inversa, el de reconocimiento de objetos y la planeación de trayectorias que involucra el control de posición. En este trabajo se propone diseñar un sistema de planeación de movimientos que aborda los subproblemas mencionados anteriormente.

1.3. Hipótesis

La planeación de movimientos para manipulación de objetos se basa en las siguientes hipótesis:

- La mejora del método de cinemática inversa con el que cuenta el robot Justina puede contribuir a mejorar algoritmos de planeación de movimientos del manipulador.
- Aplicar el análisis de componentes principales(PCA) a la nube de puntos de un objeto puede ayuar a un rápido agarre de objetos cuya forma se desconoce
- Contar con una lista de configuraciones candidatas del efector final puede ayudar al robot a hacer una mejor elección en el agarre de objetos
- Un control proporcional derivativo con compensación de gravedad puede ayudar a reducir el error de posición del manipulador robótico, por lo tanto contribuye a mejorar el agarre de objetos.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar un sistema de planeación de movimientos para manipular objetos en ambientes interiores para robots de servicio doméstico.

1.4.2. Objetivos particulares

1. Implementar el cálculo de la cinemática inversa de un manipulador de 7 grados de libertad utilizando el método de Newton-Raphson.
2. Diseñar e implementar un Control Proporcional Derivativo con compensación de gravedad para el manipulador de 7 grados de libertad.
3. Desarrollar un sistema de localización de objetos a partir de nubes de puntos.
4. Diseñar e implementar la planeación de movimientos para el agarre de objetos.

1.5. Descripción del documento

La tesis se encuentra dividida en seis capítulos.

El primer capítulo "Introducción", aborda la motivación que llevó a la realización de este trabajo, se presentan trabajos relacionados que abordaron el problema de planeación de movimientos utilizando distintas técnicas, se expone el planteamiento del problema, las hipótesis sobre las que se trabaja y finalmente los objetivos a cumplir.

El segundo capítulo "Antecedentes", trata sobre los fundamentos teóricos utilizados para cumplir los objetivos. Se incluyen definiciones de conceptos fundamentales como el de robótica y clasificación de robots, la geometría en el espacio cartesiano tridimensional y de cuerpos rígidos, pues ocupa un lugar muy importante dentro de la manipulación robótica. También se incluye el concepto de transformaciones homogéneas, indispensable para comprender como se ubica un cuerpo rígido en el espacio.

Se describen también las características de el modelo cinemático y dinámico de un manipulador robótico, esenciales para el desarrollo del controlador y para resolver el problema de la cinemática inversa. Se exponen los conceptos básicos de visión por computadora utilizados para la planeación de movimientos y se da una descripción del control propuesto para el manipulador,

así como de la plataforma en la cual se desarrolló el software y el simulador en el cual se realizaron pruebas.

El capítulo III "Visión" se describen los algoritmos utilizados para segmentar el objeto de agarre y para obtener la orientación y forma geométrica aproximada del objeto.

En el capítulo IV "Planeación de movimientos" se expone el desarrollo de la planeación de movimientos en un manipulador de 7 grados de libertad, a través de la solución numérica del problema de cinemática inversa, la propuesta geométrica para la mejor orientación del efector final basada en análisis de componentes principales, el diseño de la trayectoria y el diseño del control proporcional derivativo con compensación de gravedad.

En el capítulo V "Resultados" se muestran los resultados obtenidos en el simulador Gazebo y se contrastan con los resultados esperados. También se muestran las medidas de error del controlador.

Finalmente el capítulo VI "Discusión" trata sobre las conclusiones, observaciones sobre resultados obtenidos, así como trabajos futuros.

Capítulo 2

Marco Teórico

Este capítulo aborda conceptos y definiciones básicos de robótica, así como los fundamentos teóricos que serán utilizados en el capítulo siguiente.

2.1. Robots de servicio

La robótica es un campo relativamente nuevo, es por ello que no se cuenta con definiciones absolutas de términos como robot, robótica ó robot de servicio, en cambio existen diversas definiciones como la que proporciona la Organización Internacional para la estandarización(ISO):

Un robot de servicio es aquel robot de uso personal o profesional que realiza tareas útiles para humanos o equipos.

También define a la robótica como:

Ciencia y práctica del diseño, fabricación y aplicación de robots¹.

2.1.1. Clasificación

Las principales aplicaciones de servicio de robots de servicio son[10]:

- Paracirujanos
- Parafarmacéuticos
- De limpieza comercial
- De servicio de comida rápida
- De agricultura

¹<https://ifr.org/service-robots>

- Trabajos en el espacio
- De construcción
- Militar
- Minería
- Tareas en el hogar
- De ayuda a discapacitados y ancianos
- De vigilancia

Los robots de servicio también se pueden clasificar en robots profesionales y robots de uso personal. Los robots profesionales son aquellos Usados para una tarea comercial por personal entrenado, por ejemplo: haciendo la función de ventas y atención al cliente en supermercados y centros comerciales, sirviendo como meseros en restaurantes y cafeterías, ordenando objetos en tiendas, almacenes, bibliotecas y librerías.

Las cinco principales aplicaciones de los robots de servicio profesional registradas en 2020² son:

- Transporte y logística
- Hospitalidad y turismo
- Aplicaciones médicas
- Limpieza profesional
- Agricultura

Los robots personales son aquellos utilizado para tareas no comerciales, por ejemplo, para tareas en el hogar por personas comunes, como por ejemplo: robots que limpian y aspiran el suelo, lavan y planchan ropa, ayudan a las personas a traer y llevar objetos de un lugar a otro, o que brindan asistencia a personas de edad avanzada o enfermas³.

²<https://ifr.org/service-robots>

³<https://www.iso.org/obp/ui>

2.1.2. El manipulador robótico

La mayoría de robots de servicio están equipados con manipuladores robóticos que les permiten desempeñar tareas que requieren agarre de objetos. Un manipulador robótico es un mecanismo, que conecta cuerpos llamados eslabones a través de articulaciones, estas articulaciones pueden ser actuadores, como los motores eléctricos y generar fuerzas o pares que hacen que los eslabones del robot se muevan.

Los eslabones de un manipulador robótico se pueden organizar en serie, como el brazo de cadena abierta. Una cadena en serie es un sistema de cuerpos rígidos en el que cada miembro está conectado a otros dos, excepto el primero y el último miembros que están conectados cada uno a un solo otro miembro. Por lo general, un efector final, como una pinza o una mano para manipular objetos, se adjunta a un eslabón específico.

Los robots también están equipados con sensores para medir el movimiento en las articulaciones. Tanto para las articulaciones giratorias como para las prismáticas. La articulación giratoria (R), también llamada articulación de revolución, permite el movimiento de rotación sobre el eje de la articulación. La articulación prismática (P), permite el movimiento de traslación a lo largo de la dirección del eje de la articulación.([24]).

2.2. El problema de la planeación de movimientos para manipulación de objetos

Los robots de servicio trabajan en entornos humanos y se ven expuestos a ambientes altamente dinámicos y cambiantes. Los ambientes dinámicos implican que objetos en una determinada escena puedan cambiar de ubicación, extraerse o ingresar otros nuevos y, que factores del ambiente (iluminación, humedad, temperatura, etc) varíen continuamente. Esto eleva considerablemente la dificultad de las tareas que desempeñan en comparación a otros tipos de robots como los robots industriales, ubicados en entornos controlados, realizando tareas repetitivas y con objetos conocidos.

Por lo anterior es necesario diseñar sistemas capaces de hacer frente esas situaciones, por ejemplo si el entorno se modifica tanto que el mapa de un robot deja de ser válido y este no tuviera acceso a un nuevo mapa del entorno ni a su propia ubicación, sería conveniente que contara con un sistema de navegación que permita resolver problemas de localización y mapeo simultáneos (SLAM)[31]. En particular, para la manipulación de objetos es indispensable contar con un sistema planificador de movimientos que proporcione una solución sobre la mejor forma de tomar un objeto, el cual puede ser o no

conocido por el robot. Investigadores han planteado diversas soluciones a la manipulación de objetos con brazos robóticos aunque ninguna es definitiva y el problema sigue abierto a la investigación.

A pesar de que para un humano la acción de tomar un objeto no conlleva una gran dificultad, para un robot llevar a cabo esa sencilla acción implica la resolución de otras tareas que por su naturaleza ya son bastante complejas, como la segmentación de objetos, la cinemática directa e inversa del robot o la generación de trayectorias adecuadas para el brazo, que contengan valores de velocidad y aceleración dentro de rangos establecidos para los actuadores. También es necesario contar con un sistema de visión (proporcionado por una cámara de profundidad), ya que permite recabar información de los objetos en la escena, como su color, cantidad, dimensiones, posición y orientación espacial.

Un sistema planificador de movimientos se encarga de realizar la planeación de configuraciones que debe seguir un manipulador para tomar de manera eficiente un objeto del ambiente, gestiona el orden en que han de realizarse las tareas que llevan a una solución del problema principal, y asegura mediante un controlador que el brazo robótico vaya de un estado inicial a un estado final apropiado para la toma de objetos, es decir, de la planeación de movimientos deben derivar acciones concretas ejecutables por el robot de manera óptima. El control de movimientos del efector final demanda un análisis preciso de las características de la estructura mecánica, actuadores y sensores, de este análisis derivan modelos matemáticos de las componentes del robot [22].

Debido a la practicidad que brinda el control en el espacio articular es necesario proporcionar al planificador las funciones de cinemática directa e inversa para poder pasar del espacio de posiciones articulares q al espacio cartesiano (x, y, z) y viceversa, las cuales dependen directamente de la geometría del robot.

En este trabajo se propone un sistema planificador de movimientos que brinde configuraciones óptimas para el agarre de objetos desconocidos. Mediante el procesamiento de los datos recabados por una cámara de profundidad (RGB-D) se obtendrá información del objeto que será analizada con métodos estadísticos que estimen la configuración espacial del objeto así como sus dimensiones, para posteriormente generar un conjunto de configuraciones candidatas del efector final a partir del análisis geométrico y se identificara mediante ciertos criterios la mejor del conjunto propuesto.

Para alcanzar el objetivo de desarrollar un sistema de planeación de movimientos se debe alcanzar los siguientes objetivos particulares

- Creación de un algoritmo para reconocimiento y ubicación de objetos:

El primer paso para manipular un objeto es reconocerlo y ubicar su posición en el ambiente.

- Creación de un algoritmo para estimar posición y orientación de objetos detectados: Mediante el análisis de componentes principales (PCA) es posible estimar la orientación y dimensiones de un objeto en la escena.
- Creación de un algoritmo para el cálculo de configuraciones óptimas del efector final: Haciendo un análisis geométrico se consigue la mejor pose del efector final(EF) para tomar un objeto.
- Implementación de mejoras al algoritmo de cinemática inversa actual del robot Justina: El desarrollo de un algoritmo para generar trayectorias conseguirá soluciones para puntos distantes del punto inicial en el algoritmo Newton-Raphson utilizado para resolver la cinemática inversa.
- Diseño de un controlador proporcional derivativo con compensación de gravedad: un control de posición realimentado es necesario, ya que asegura que cada una de las articulaciones del brazo lleguen a la posición deseada.

2.3. Trabajos relacionados

El problema de la manipulación de objetos comienza con la identificación del objeto deseado. La cámara RGB-D situada en la cabeza del robot proporciona una nube de puntos del entorno, existen múltiples soluciones al problema de aislar la nube de puntos del objeto que se desea tomar. Algunos autores[15] proponen como primer paso para la segmentación del objeto, filtrar la nube de puntos de vista parcial del objeto para eliminar los puntos más distantes, asumen que los objetos se encuentran sobre una superficie plana y utilizan el consenso RANSAC para localizar la mesa y mantener solo la información de los objetos sobre ella, finalmente aplican segmentación Euclidiana para obtener la nube de puntos de cada objeto, mientras que en este trabajo se realizó una segmentación por contornos.

Una vez aislada la nube de puntos, es procesada para obtener información del objeto desconocido que es necesaria para generar una solución de agarre. Este problema suele ser abordado desde múltiples enfoques, uno de estos enfoques recurre a las redes neuronales. D.R.Faria[11] resuelve el problema de manipulación de objetos implementando un sistema basado en el conocimiento de demostraciones de agarre de objetos realizados por humanos, este sistema consta de una etapa de aprendizaje, en la cual, a partir de

las lecturas de sensores colocados en una mano humana, son etiquetados los puntos de contacto y las regiones agarrables de objetos diversos. La forma del objeto Q es estimada a través de primitivas geométricas. Esta información es procesada con un enfoque probabilístico, pues se establece $Q \rightarrow$ objeto y $G \rightarrow$ todas las formas posibles de agarre como variables aleatorias y se define la dependencia de una con respecto a la otra mediante una distribución de probabilidad condicional $P(Q|G)$.

Después de establecer la mejor forma de agarre del objeto, son almacenadas en una base de datos una cantidad de primitivas geométricas asociadas con un tipo de agarre.

En la segunda etapa, con la información del objeto que se desea manipular y que es obtenida con una cámara RGB-D, es utilizada para aproximar la forma del objeto con primitivas geométricas y comparada con las que hay en la base de datos para obtener la forma de manipularlo.

P.Rivera[28] propone para conseguir poses de agarre similares a las de una mano humana, entrenar una red neuronal con cuatro capas de 256 neuronas, usando aprendizaje de refuerzo profundo, introduciendo un espacio de sinergia que influye en el aprendizaje con el fin de que el robot aprenda poses de mano óptima para la manipulación natural de los objetos de formas diversas a partir de información poses de manos obtenidas de demostraciones de agarre humano. se trabaja con una mano robótica antropomórfica e información proveniente de nubes de puntos hápticas.

Dentro del mismo enfoque, D.Rodríguez[29] propone dividir el problema del agarre de objetos en una fase de aprendizaje y una fase de inferencia. Bajo la suposición de que los objetos dentro de una categoría son similares en forma y uso, en la fase de aprendizaje, el modelo lineal que representa a la categoría se aprende seleccionando una única instancia del conjunto de datos de entrenamiento para que sea el modelo canónico de la categoría. Luego, se encuentran las transformaciones que relacionan esta instancia con todas las demás instancias de la categoría. Para cada instancia del conjunto de entrenamiento, un descriptor de agarre asociado. De esta manera, se pueden agregar múltiples experiencias en el modelo canónico. En la fase de inferencia, dada una nueva instancia observada, el método busca en el subespacio de transformaciones la transformación que mejor relaciona la forma canónica con la instancia observada. Dependiendo de las variables de forma latente resultantes y del conocimiento de agarre agregado acumulado en el modelo canónico, se infiere un descriptor de agarre para la instancia novedosa.

Las redes neuronales requieren una gran cantidad de datos para su funcionamiento y en la mayoría de los casos también requieren mucho tiempo de entrenamiento y procesadores costosos. Un enfoque alternativo consiste en el análisis geométrico de la nube de puntos del objeto. B.Zapata [34])

propone un método para encontrar el mejor par de puntos de agarre basado en un conjunto de reglas geométricas. Consistente en aproximar el centroide del objeto, así como su eje principal y su orientación utilizando el análisis de componentes principales. Se establecen regiones de agarre sobre la superficie del objeto, en dirección perpendicular al eje principal y cercanos al centroide.

Para aumentar las probabilidades de que un agarre sea estable se priorizan las regiones planas y se evitan aquellas con un mayor grado de curvatura. Un algoritmo mide la variación de posición entre un punto y sus vecinos para obtener la curvatura. Los puntos elegidos p_1 y p_2 de agarre tienen menor curvatura que el resto. Este método utiliza una cámara RGB-D externa al robot y fija en un sitio.

Otro artículo con el mismo enfoque[15] propone un sistema de agarre de objetos para la generación de poses del efector final, de acuerdo con un cuadro delimitador orientado(OBB). Define dos posibles tipos de agarre para el efector final: agarre superior o agarre lateral. Aplica el análisis de componentes principales para aproximar la forma y la pose del objeto. procede a buscar puntos de contacto sobre el lado más largo de un objeto, sobre los agarres plantea que si la altura es adecuada, es posible utilizar un movimiento de agarre lateral, de lo contrario, realiza un movimiento de agarre superior.

Q.Lei[21] propone utilizar el análisis de componentes principales sobre la nube de puntos del objeto, y a partir del eje con mayor dimensión construir el sistema coordenado asociado. Para abordar la incertidumbre de la información obtenida de la nube de puntos del objeto derivada de la oclusión, propone desplazar al robot a otro punto del lugar para conseguir una perspectiva distinta del objeto, o bien, utilizar dos cámaras de profundidad, localizadas en la mano y cabeza del robot, y con las dos nubes de puntos construir una "gran" nube de puntos con que se pueda realizar una reconstrucción del objeto.

El grupo de agarres candidatos contiene a los agarres generados por la rotación del sistema coordenado del efector final sobre el eje principal, desde la parte superior de la nube de puntos hasta la parte inferior. Se elige al candidato que requiere la mínima fuerza de agarre, esta información se consigue a partir del análisis del balance de fuerzas sobre la nube de puntos del objeto.

El método propuesto por L. Sainul[30] busca puntos de contacto del efector final con el objeto que se desea manipular, utilizando el análisis de componentes principales y medidas de la calidad de agarre. El enfoque de análisis de componentes principales es aplicado en los vértices de los objetos para alinear la pinza con el objeto. Luego, se genera un conjunto de agarres y se aplica un método de corte de objetos para encontrar rápidamente puntos de contacto en el objeto y evaluar la calidad del agarre.

La información de la forma del objeto se utiliza para clasificar los objetos

en varias categorías. Si el objeto es de tipo cilíndrico/barra (objeto unidimensional más grande) es realizado un agarre lateral, si el objeto es de tipo esférico/cuboide (objeto tridimensional más grande) se realiza un agarre envolvente, y para objetos planos (objetos bidimensionales) se realiza un agarre desde los bordes.

2.3.1. *MoveIt!*

Se trata de un paquete de software de código abierto, diseñado para la plataforma ROS(Robot Operating System). Su propósito principal consiste en brindar las trayectorias necesarias para que el brazo de un robot pueda llevar el efector final a una posición determinada.

Contiene librerías de funciones que proporcionan diferentes tipos de agarre robótico; geométricos o por aprendizaje de máquina, así como también contiene nodos que permiten llevar a cabo planificación de trayectorias percepción 3D, cálculos cinemáticos, control de colisiones y navegación ⁴.

MoveIt! utiliza una arquitectura basada en complementos para resolver la cinemática inversa al tiempo que proporciona una implementación nativa de la cinemática directa. De forma nativa, MoveIt! utiliza un solucionador numérico para cinemática inversa para cualquier robot. Los usuarios pueden agregar sus propios solucionadores personalizados, en particular, los solucionadores analíticos son mucho más rápidos que el solucionador nativo.[18].

2.4. Espacio de configuración

La configuración de un cuerpo rígido es una especificación de la ubicación de todos sus puntos. Para un cuerpo rígido que se mueve en un espacio tridimensional, se necesitan seis parámetros independientes para especificar la configuración. La configuración de un robot es una especificación de la configuración de todos sus eslabones. El espacio de configuración del robot es el conjunto de todas las configuraciones posibles del robot. La dimensión del espacio de configuración C es el número de grados de libertad de un robot.

2.4.1. Articulaciones robóticas

Un manipulador robótico es un mecanismo, que conecta cuerpos llamados eslabones a través de articulaciones, estas articulaciones pueden ser actuadores, como los motores eléctricos y generar fuerzas o pares que hacen que los

⁴<https://www.theconstructsim.com/ros-moveit/>

eslabones del robot se muevan. Los eslabones de un manipulador robótico se pueden organizar en serie, como el brazo de cadena abierta.

Por lo general, un efector final, como una pinza o una mano para manipular objetos, se adjunta a un eslabón específico. Los robots también están equipados con sensores para medir el movimiento en las articulaciones. Tanto para las articulaciones giratorias como para las prismáticas. La articulación giratoria (R), también llamada articulación de revolución, permite el movimiento de rotación sobre el eje de la articulación. La articulación prismática (P), permite el movimiento de traslación a lo largo de la dirección del eje de la articulación. ([24]).

2.4.2. Espacio de tareas y espacio de trabajo

El espacio de tareas de un robot es un espacio en el que la tarea del robot se puede expresar de forma natural. El espacio de trabajo de un robot es una especificación de las configuraciones que puede alcanzar el efector final del robot.

2.5. Representación de posición y orientación

La posición y orientación de un cuerpo rígido en el espacio se denomina colectivamente pose. Una posible representación de la pose de un objeto es un sistema coordenado. Un sistema de coordenadas i consta de un origen, denominado O_i y una triada de vectores de base mutuamente ortogonales, denominados $(\hat{x}_i, \hat{y}_j, \hat{z}_k)$, que están todos fijos dentro de un cuerpo particular. La pose de un cuerpo siempre se expresará en relación con algún otro cuerpo, por lo que puede expresarse como la pose de un sistema coordenado relativo a otro. [3]

2.5.1. Posición y desplazamiento

La posición del origen del sistema coordenado i relativo al sistema coordenado j puede ser especificada por el vector de 3x1

$$p_i^j = \begin{pmatrix} p_{ix}^j \\ p_{iy}^j \\ p_{iz}^j \end{pmatrix} \quad (2.1)$$

Las componentes de este vector son las coordenadas cartesianas del origen O_i en el sistema j , y que corresponden a las proyección del vector p_i^j sobre los ejes correspondientes.

Una traslación es un desplazamiento en el cual ningún punto del cuerpo rígido permanece en su posición inicial.

2.5.2. Orientación y rotación

Una rotación es un desplazamiento en el cual al menos un punto del cuerpo rígido permanece en su posición inicial.

Matrices de Rotación

La orientación del sistema coordenado i relativo al sistema coordenado j se puede denotar expresando los vectores base $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$ en términos de los vectores base $(\hat{x}_j, \hat{y}_j, \hat{z}_j)$. Esto produce $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$, que cuando se escriben juntos como una matriz de 3×3 se conoce como matriz de rotación. Los componentes de jR_i son los productos punto de los vectores base de los dos marcos de coordenadas

$$R_i^j = \begin{pmatrix} \hat{x}_i \cdot \hat{x}_j & \hat{y}_i \cdot \hat{x}_j & \hat{z}_i \cdot \hat{x}_j \\ \hat{x}_i \cdot \hat{y}_j & \hat{y}_i \cdot \hat{y}_j & \hat{z}_i \cdot \hat{y}_j \\ \hat{x}_i \cdot \hat{z}_j & \hat{y}_i \cdot \hat{z}_j & \hat{z}_i \cdot \hat{z}_j \end{pmatrix} \quad (2.2)$$

Debido a que los vectores base son vectores unitarios y el producto punto de cualquiera vector unitario es el coseno del ángulo entre ellos, las componentes son comúnmente referidas como cosenos directores.

Una rotación elemental de un sistema coordenado i alrededor del eje \hat{z}_j una cantidad de ángulos θ se puede representar con la siguiente matriz

$$R_z(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (2.3)$$

mientras que la misma rotación sobre el eje \hat{y}_j es

$$R_y(\theta) = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \quad (2.4)$$

y sobre el eje \hat{x}_j es

$$R_x(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{pmatrix} \quad (2.5)$$

Las matrices de rotación pueden ser combinadas utilizando multiplicación de matrices para expresar la orientación del sistema coordenado i con respecto al sistema coordenado k [3].

$$R_i^k = R_j^k R_i^j \quad (2.6)$$

Propiedades de las matrices de rotación

- $R \in SO(n)$
- $R^{-1} \in SO(n)$
- $R^{-1} = R^T$
- Cada columna (y por lo tanto cada renglón) de R es un vector unitario.
- Las columnas (y por lo tanto los renglones) de R son mutuamente ortogonales.
- $\det\{R\} = 1$

2.5.3. Composición de rotaciones

Se puede aplicar una secuencia de rotaciones a un objeto; se tienen dos opciones: Rotación con respecto al marco de coordenadas actual (intrínseco) Rotación con respecto a un marco fijo (extrínseco).

Rotaciones extrínsecas

Si se realiza pos-multiplicación, la rotación es intrínseca, se aplica al sistema de coordenadas actual.

Rotaciones intrínsecas

Si se realiza premultiplicación, la rotación es extrínseca y se aplica al sistema de coordenadas fijo [25].

2.5.4. Parametrización de rotaciones

Un cuerpo rígido tiene solo tres grados de libertad de rotación, por lo tanto, solo se requieren tres valores para especificar su orientación. Existen varios métodos para especificar una rotación con ángulos, que hacen más fácil la visualización de las rotaciones a los seres humanos. Están los ángulos de Euler, la representación eje-ángulo o los ángulos roll,pitch,yaw.

Ángulos Roll, Pitch, Yaw (RPY)

Una matriz de rotación también puede ser descrita como un producto de rotaciones sucesivas alrededor de los ejes del sistema coordinado. Estas rotaciones están definidas por los ángulos *roll*, *pitch* y *yaw*[14].

El orden de rotación es x-y-z, primero un *yaw* alrededor del eje x por un ángulo ψ , luego un *pitch* alrededor del eje y por un ángulo θ , y finalmente un *roll* sobre el eje z en un ángulo ϕ . Dado que las rotaciones sucesivas son relativas al marco de coordenadas universales, la matriz de rotación resultante está dada por

$$\begin{aligned}
 R &= R_z(\phi)R_y(\theta)R_x(\psi) \\
 &= \begin{pmatrix} C_\phi & -S_\phi & 0 \\ S_\phi & C_\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} C_\theta & 0 & S_\theta \\ 0 & 1 & 0 \\ -S_\theta & 0 & C_\theta \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & C_\psi & -S_\psi \\ 0 & S_\psi & C_\psi \end{pmatrix} \\
 &= \begin{pmatrix} C_\phi C_\theta & -S_\phi C_\psi + C_\phi S_\theta S_\psi & S_\phi S_\theta + C_\phi S_\psi C_\psi \\ S_\phi C_\theta & C_\phi C_\psi + S_\phi S_\theta S_\psi & -C_\phi S_\psi + S_\phi S_\theta C_\psi \\ -S_\theta & C_\theta S_\psi & C_\theta C_\psi \end{pmatrix}; \quad (2.7)
 \end{aligned}$$

2.5.5. Transformaciones homogéneas

Un robot puede ser colocado en cualquier posición y orientación realizando una rotación seguida de una traslación. Combinando estas dos operaciones en una sola matriz se obtiene una matriz conocida como matriz de transformaciones homogéneas.

Cualquier vector r^i expresado en coordenadas del sistema i puede ser expresado en coordenadas del sistema j si se conoce la posición y orientación del sistema i con respecto al sistema j . La posición del origen del sistema coordinado i relativo al sistema coordinado j puede ser expresada a través del vector $p_i^j = (p_{ix}^j, p_{iy}^j, p_{iz}^j)$. La orientación del sistema coordinado i relativo al sistema j puede ser expresada con la matriz de rotación R_i^j , por lo tanto

$$r^j = R_i^j r^i + p_i^j \quad (2.8)$$

Esta ecuación puede ser escrita como

$$\begin{pmatrix} r^j \\ 1 \end{pmatrix} = \begin{pmatrix} R_i^j & p_i^j \\ 0^T & 1 \end{pmatrix} \begin{pmatrix} r^i \\ 1 \end{pmatrix} \quad (2.9)$$

donde

$$T_i^j = \begin{pmatrix} R_i^j & p_i^j \\ 0^T & 1 \end{pmatrix} \quad (2.10)$$

Es la matriz de transformaciones homogéneas de 4x4 y p_i^j es la representación homogénea del vector de posición r^j y r^i . La matriz T_i^j transforma vectores desde el sistema coordenado i al sistema coordenado j . Su transformación inversa T^{-1} , transforma vectores desde el sistema coordenado j al sistema coordenado i . La composición de matrices de transformación homogéneas de 4x4, se logra mediante la simple multiplicación de matrices, al igual que en el caso de matrices de rotación de 3x3.

$$\text{inv}(T_i^j) = \begin{pmatrix} R_i^{jT} & -R_i^{jT} p_i^j \\ 0^T & 1 \end{pmatrix} \quad (2.11)$$

Las transformaciones homogéneas son particularmente útiles cuando se desea una notación compacta. Sin embargo, esta no es una representación computacionalmente eficiente ya que introduce una gran cantidad de multiplicaciones adicionales por unos y ceros[3].

2.5.6. Árbol de transformaciones homogéneas

En los robots suelen asignarse los sistemas coordenados sobre los actuadores y cada uno está relacionado con el inmediato anterior. Es posible conocer la posición del efector final en coordenadas del sistema base realizando la transformación de coordenadas de un sistema a otro. Una estructura en la que cada sistema coordenado se define por su relación con otro (y solo uno) se conoce como árbol de transformaciones y define las relaciones entre diferentes sistemas de coordenadas, en términos de traslación, rotación y movimiento relativo. Permite convertir un punto conocido en un sistema coordenado en cualquier otro sistema en el árbol.

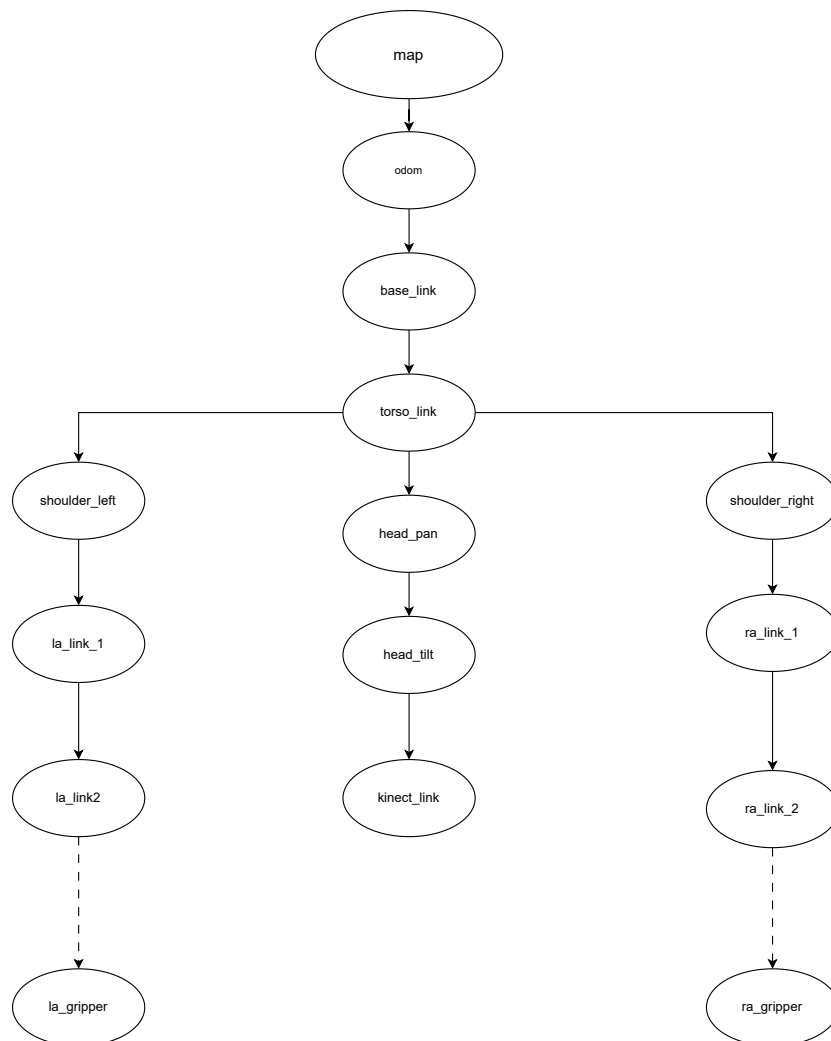


Figura 2.1: Árbol de transformaciones del robot utilizado en este trabajo.

En este capítulo se dio una breve introducción a los conceptos relacionados con la robótica de servicio, se describió el problema de la planeación de movimientos, y se dio un resumen de los trabajos relacionados que han abordado el problema desde distintos enfoques. Se presentan conceptos que frecuentemente serán mencionados en este trabajo como el espacio de trabajo del manipulador, posición y orientación, matriz de rotación y transformaciones homogéneas. Los cuales son de gran importancia pues son la base del conocimiento para diseñar del planeador de movimientos y para comprender el trabajo realizado posteriormente.

En los siguientes capítulos, visión y planeación de movimientos, se aborda el trabajo desarrollado desde que el robot tiene que reconocer un objeto del

entorno, hasta el punto en el cual elige la mejor configuración del manipulador para tomar ese objeto.

Capítulo 3

Detección y localización de objetos

Los fenómenos de visión y perspectiva en 3D fueron estudiados por primera vez por los arquitectos y artistas de la antigua Grecia. Posteriormente, muchos pintores aplicaron técnicas de perspectiva para producir la ilusión de profundidad en las pinturas planas. A Leonardo da Vinci se le atribuye la invención de la cámara oscura, un prototipo de las cámaras modernas, que ayudó a adquirir imágenes de una escena 3D en un plano. Luego, la Óptica de Isaac Newton sentó las bases de la ciencia de la visión[7].

La visión por computadora es la ciencia de crear una capacidad de visión, similar a la de los humanos, en las computadoras. Un sistema de visión es algo que evalúa datos de una fuente de imagen (típicamente una cámara), extrae datos sobre esas imágenes y hace algo con los resultados. Un sistema de visión típico realiza dos pasos generales. El primer paso es filtrar la entrada para reducir el rango de información a procesar. El segundo paso es extraer y procesar las características clave de la(s) imagen(es)[19]. Algunas características de uso común incluyen:

- Información de color: buscando cambios de color para detectar objetos.
- Extracción de blobs: detección de píxeles adyacentes de colores similares.
- Bordos y esquinas: examen de los cambios de brillo para identificar los bordes de los objetos.
- Reconocimiento de patrones y coincidencia de plantillas: agregar inteligencia básica al hacer coincidir las características con las características de los objetos conocidos.

En el campo de visión por computadora es fundamental representar la posición y orientación de objetos en el entorno. Para poder extraer esta información es necesario aislar la nube de puntos del objeto del conjunto de puntos del entorno, para lograrlo existen varias técnicas: segmentación por distancia, por color, por contornos, etc. En las siguientes secciones se describe el método implementado para la detección de contornos de los objetos en la escena.

Usando una cámara de profundidad se obtiene una nube de puntos y una imagen RGB, la imagen RGB es cuantizada por color con el algoritmo k-medias para simplificar la detección de contornos, y la nube de puntos es utilizada para el análisis de componentes principales y consecuentemente estimar la orientación y el centroide del objeto, y así poder construir un sistema coordinado asociado al objeto.



Figura 3.1: Entornos comunes con los que suele encontrarse un robot de servicio.

En este trabajo se utiliza una nube de puntos con vista parcial de los objetos en la escena para desempeñar agarres de objetos

3.1. Adquisición de datos con *Kinect*

Kinect es una herramienta flexible que puede ser utilizada en aplicaciones de diversas áreas como: Computación Gráfica, Procesamiento de Imágenes, Visión por Computador e Interacción Hombre-Máquina. Los datos adquiridos

tienen naturalezas diferentes y complementarias, combinando geometría con atributos visuales.

Cuenta con una cámara RGB y una cámara de infrarrojos (IR). Son capaces de capturar una imagen en color y la profundidad de cada píxel de la escena. Estos datos contienen información visual y geométrica de la escena. La mayoría de los sistemas de procesamiento de imágenes se basan únicamente en los canales de color de las imágenes. Sin embargo, otros atributos de la imagen pueden ser utilizados para el procesamiento, por ejemplo: profundidad, normal, luminancia, etc. Los atributos contienen información adicional que permite implementar algunos procedimientos que son difíciles, cuando no imposibles, usando solo colores.

La representación de datos más natural que se obtiene con Kinect es una Imagen RGB-D, es decir, una combinación de tres canales de color (uno para cada color: rojo, verde y azul) y otro para los datos de profundidad. Los canales de color se pueden representar mediante una matriz de enteros de 8 bits (capaz de representar 256 colores cada uno). Los datos de profundidad se pueden representar mediante una matriz de enteros de 16 bits. La representación del dispositivo de profundidad utiliza un número entero de 16 bits, pero transfiere los datos con números enteros de 11 bits para ahorrar espacio[6].



Figura 3.2: Sistema Kinect montado en la cabeza del robot Justina.

3.2. Segmentación de imagen

El propósito de este proceso es muy parecido a la idea de separar la figura del fondo, siendo la idea, dividir la imagen en regiones que sean significativas para el propósito en cuestión[26].

Las regiones pueden ser definidas como grupos de píxeles que deben ser homogéneos con respecto a alguna característica, tal como color, textura, profundidad, entre otras. Cuando el primer plano es de interés el fondo suele ser ignorado.

La segmentación tiene dos objetivos, el primero es descomponer la imagen en partes para su análisis futuro. El segundo objetivo de la segmentación es realizar un cambio de representación. Los píxeles de la imagen deben ser organizados en unidades de más alto nivel más eficientes para análisis.[32]

3.2.1. Segmentación por agrupamiento de píxeles en base al color

El agrupamiento es el proceso por el cual un conjunto de datos es reemplazado por grupos, los cuales son colecciones de datos de puntos de datos que permanecen juntos. El criterio de agrupamiento dependerá de la aplicación; los píxeles podrían agruparse por cercanía, textura, color, etc.

El procedimiento para segmentar una imagen por agrupamiento comienza cuando se representa cada píxel de la imagen con un vector de características. Este vector de características contiene todas las medidas que podrían ser relevantes en la descripción del píxel. Entre las características naturales de los vectores por las cuales se puede realizar el agrupamiento son: la intensidad del píxel, el color, su localización, textura. Cada vector de características pertenece a exactamente un agrupamiento, y cada agrupamiento representa un segmento de la imagen. [8]

3.2.2. Agrupamiento por color usando el algoritmo *k-Medias*

El algoritmo K-medias es un método simple e iterativo, se muestra a continuación.

Este algoritmo funciona dividiendo un conjunto de elementos de entrada en k grupos, se asume que cada grupo tiene un centroide. Inicialmente los centroides se colocan en posiciones aleatorias, y el algoritmo de forma iterativa, calcula la distancia entre datos y centroides, asociando cada dato con el centroide mas cercano, hasta converger a un número fijo de iteraciones.

Algorithm 1 Agrupamiento iterativo con k-medias

-
- 1: **for** m K-medias agrupamientos de un conjunto de vectores $x_i = (1, 2, 3, \dots, n)$ **do**
 - 2: Establecer contador a 1 ($i_c = 1$)
 - 3: Elegir aleatoriamente un conjunto de k medias $m_1(1), m_2(1), \dots, m_k(1)$.
 - 4: **for** cada vector x_i **do**
 - 5: Calcula $D(x_i, m_k(i_c))$ para cada $k = 1, \dots, k$ y asigna x_i al agrupamiento C_j con el mean mas cercano.
 - 6: Incrementa i_c en 1 y actualiza las medias para obtener un nuevo conjunto $m_1(i_c), m_2(i_c), \dots, m_k(i_c)$.
 - 7: **end for**
 - 8: **end for**
 - 9: Repetir hasta que $C_k(i_c) = C_k(i_c + 1)$ para toda k.
-

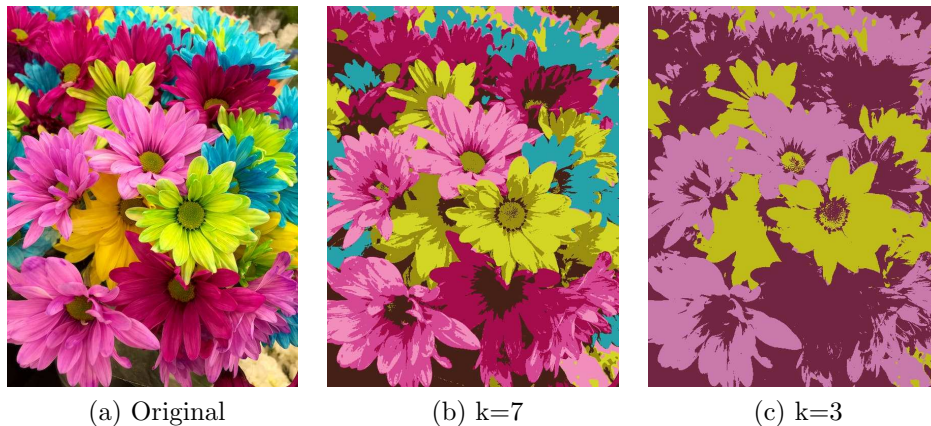


Figura 3.3: Aplicación del algoritmo de agrupamiento k-Medias en un espacio RGB, para diferentes valores del parámetro k, mientras mas alto sea, la cantidad de colores en la imagen es mayor.

3.3. Detección de objetos en la escena

Partiendo de la suposición de que es posible detectar el contorno de objetos que se encuentran sobre una mesa, es posible utilizar una imagen RGB-D de la escena tomada con la cámara de profundidad del robot. Las imágenes de la escena suelen contener una amplia gama de colores debido a la variedad de objetos, y esto dificulta la tarea de obtener el contorno de los objetos a manipular.



Figura 3.4: Imagen captada por la cámara RGB-D en simulación.

Para facilitar la detección de contornos de objetos debe realizarse una segmentación por color de los píxeles de la imagen, para la cual es necesario procesar la imagen a través de una técnica conocida como cuantización del color, la cual reduce el número de colores en la imagen. La cuantización por color se consigue realizando un agrupamiento por color con el algoritmo de k-medias presentado anteriormente.

3.3.1. Pre-procesamiento de la imagen

La nube de puntos sin procesar suele ser densa, y puede contener ruido, esto afecta al desempeño en el reconocimiento de objetos. Suponiendo que los objetos que deben ser manipulados se encuentran sobre una mesa, es posible filtrar la nube mediante un extractor de plano que encuentra el plano que mejor represente a la mesa y elimina los puntos por debajo de él (fig 4.15a).

3.3.2. Cuantización por color de la imagen

Para realizar la cuantización por color es necesario redimensionar la imagen RGB, (ya que contiene 3 características R,G,B), a una matriz de tamaño $M \times 3$ (M es el número de píxeles en la imagen). Se establece el criterio de convergencia de los centroides para el algoritmo. la imagen cuantizada se muestra en la figura 4.16c.

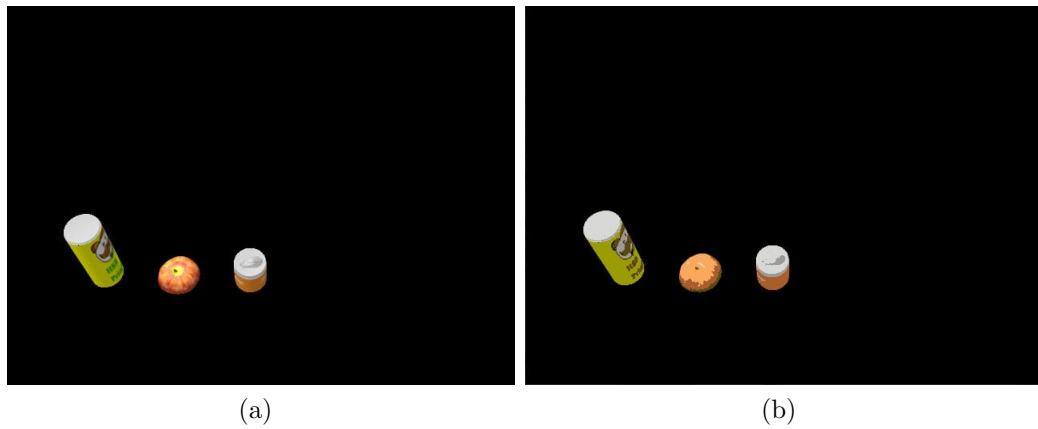
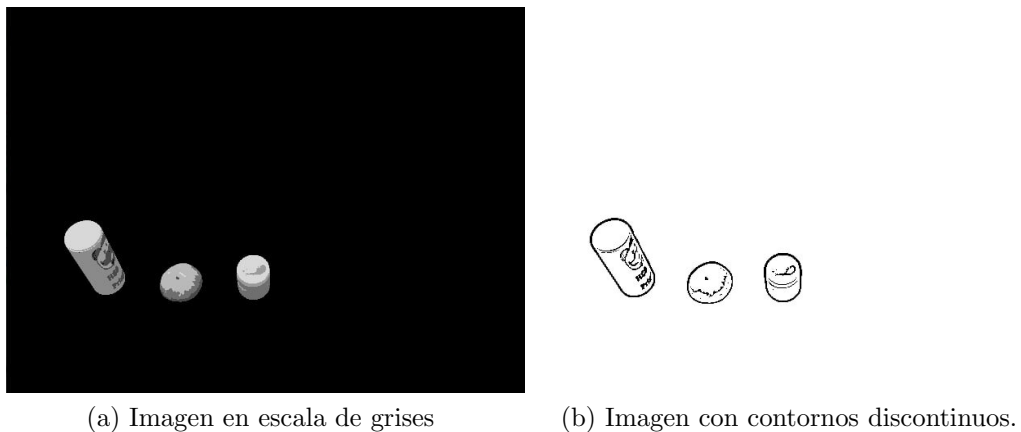


Figura 3.5: (a) Filtrado de la nube de puntos con un extractor de plano. (b) Imagen cuantizada en color.

3.3.3. Detección de contornos

La imagen cuantizada en color, debe ser convertida a escala de grises para facilitar el proceso siguiente, esta acción además reduce el coste computacional. La umbralización de la imagen permite binarizarla (fig3.6a). OpenCV contiene funciones que permiten cambiar entre espacios de color y binarizar imágenes.

El resultado es una imagen binarizada con objetos encerrados por contornos discontinuos (fig3.6b).



(a) Imagen en escala de grises

(b) Imagen con contornos discontinuos.

Figura 3.6: .

Para mejorar las probabilidades en la detección de contornos es necesario

realizar transformaciones morfológicas sobre la imagen binaria, con el fin de conectar todos los posibles contornos que, al umbralizar la imagen quedan discontinuos debido a la variedad de grises presentes en el objeto(fig3.6b). La transformación utilizada para este propósito son la dilatación y erosión. Estas operaciones tienden a preservar el área de las regiones conectadas con mayor precisión).

La dilatación es una convolución de alguna imagen (o región de una imagen) A, con algún núcleo B. El núcleo puede considerarse como una plantilla o máscara, y su efecto para la dilatación es el de un operador máximo local. A medida que el núcleo B se escanea sobre la imagen, se calcula el valor máximo de píxel superpuesto por B y se reemplaza el píxel de la imagen debajo del punto de anclaje con ese valor máximo.

La operación de dilatación se utiliza a menudo cuando se intenta encontrar componentes conectados (es decir, grandes regiones discretas de color o intensidad de píxel similar). La utilidad de la dilatación surge porque en muchos casos una región grande podría dividirse en múltiples componentes como resultado del ruido, las sombras o algún otro efecto similar. Una pequeña dilatación hará que dichos componentes se "fundan" en uno solo.(3.7a).

La erosión es la operación inversa. La acción del operador de erosión equivale a calcular un mínimo local sobre el área del núcleo. La operación de erosión se utiliza a menudo para eliminar el ruido "moteado" en una imagen3.7b. La idea aquí es que las motas se erosionen hasta quedar en nada, mientras que las regiones más grandes que contienen contenido visualmente significativo no se ven afectadas.[12].

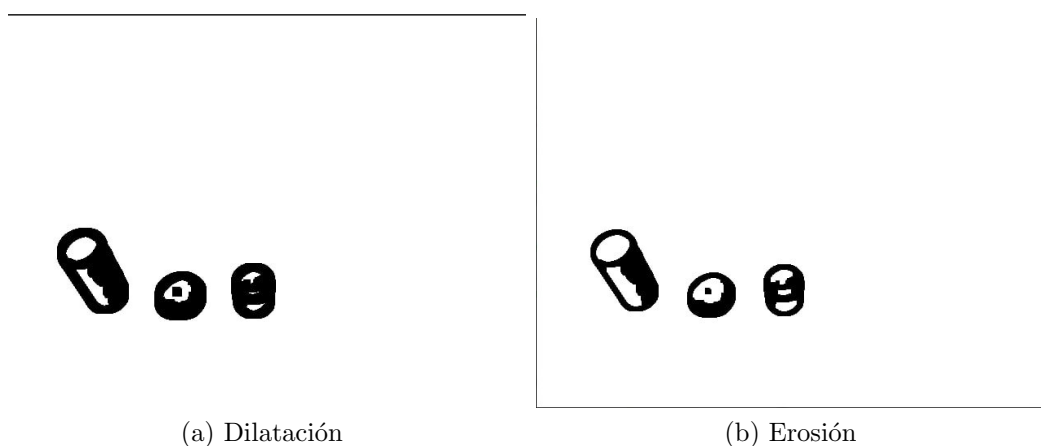


Figura 3.7: Procesamiento de la imagen.

Los contornos se representan en OpenCV mediante secuencias en las que

cada entrada de la secuencia codifica información sobre la ubicación del siguiente punto de la curva.

OpenCV cuenta con bibliotecas de funciones que permiten obtener contornos de imágenes. La función *cvFindContours()* calcula contornos a partir de imágenes binarias.

cv2.findContours(imagen, modo, método)

- **Imagen:** Una imagen binaria es aquella en la cual todos los píxeles que no son cero, se consideran como '1'. En otras palabras, si el pixel no esta apagado se considera como color blanco.
- **Modo:** Modo de recuperación del contorno, se optó por recuperar todos los contornos con sus jerarquías. La jerarquía demuestra la relación que tienen los contornos, por ejemplo, un contorno dentro de otro, el contorno externo será padre, mientras que el interno el hijo.
- **Método(Método de aproximación):** Parámetro que indica cómo se quiere aproximar el contorno. Se puede almacenar todos los puntos del contorno o solo aquellos que no son puntos redundantes.

Argumentos de salida:

- **Contornos:** Contornos detectados, almacenados como vectores de puntos.
- **Jerarquía:** Entrega una lista con la relación entre los contornos encontrados.

Finalmente se aplica una mascara de bits al contorno del objeto a manipular para extraer su nube de puntos, el resultado se muestra en la figura 3.8b. La figura 5.18 muestra un diagrama de bloques del algoritmo utilizado para la detección de objetos.

3.4. Orientación mediante PCA

3.4.1. Estimación de la posición del objeto

Una vez que se tiene la nube de puntos del objeto aislado es posible calcular el centroide del objeto, el cual servirá mas adelante para usarlo como origen del sistema coordenado del objeto. Sea la nube de puntos del objeto, que es un subconjunto de la nube de puntos obtenida con la cámara de profundidad $\mathbb{D} \subset \mathbb{C}$. Para obtener el centroide:

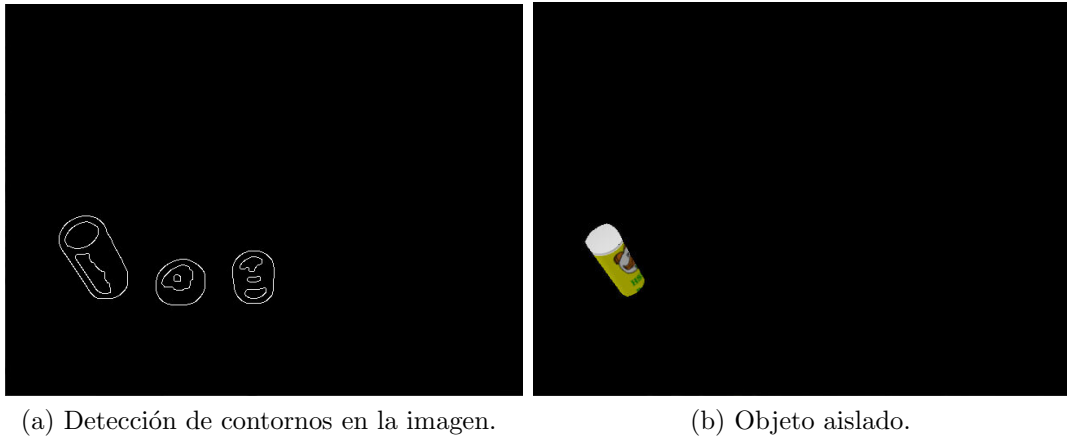


Figura 3.8

$$C_{obj}(x, y, z) = \left(\frac{1}{n} \sum_{i=1}^n px_i, \frac{1}{n} \sum_{i=1}^n py_i, \frac{1}{n} \sum_{i=1}^n pz_i \right) \quad (3.1)$$

Donde:

$$(p_x, p_y, p_z) = p \in \mathbb{D}$$

$$i = 1, 2, 3, \dots$$

n = número de puntos de la nube del objeto.

3.4.2. Estimación de la orientación del objeto

El análisis de componentes principales (PCA) es una técnica estadística que se aplica a un conjunto de datos con el objetivo de extraer las características más importantes. Esto se logra transformando a un nuevo conjunto de variables, los componentes principales, que no están correlacionados y que están ordenados de modo que los primeros retengan la mayor parte de la variación presente en todas las variables originales[16]. Estos datos pueden ser cualquier tipo; temperaturas, edades, pesos, distancias, etc. Para este trabajo el conjunto de datos sobre el que se aplica el PCA es el conjunto de puntos de coordenadas que conforman a la nube de puntos del objeto que se desea manipular.

Con las componentes principales de ese conjunto de puntos es posible reconstruir una estimación de sus dimensiones y un sistema coordenado asociado al objeto que, brinde información sobre su pose con respecto a otro sistema coordenado del robot. Estas componentes ayudan a determinar cuál de los ejes del objeto es más fácil de agarrar. Si se observa la forma en que los

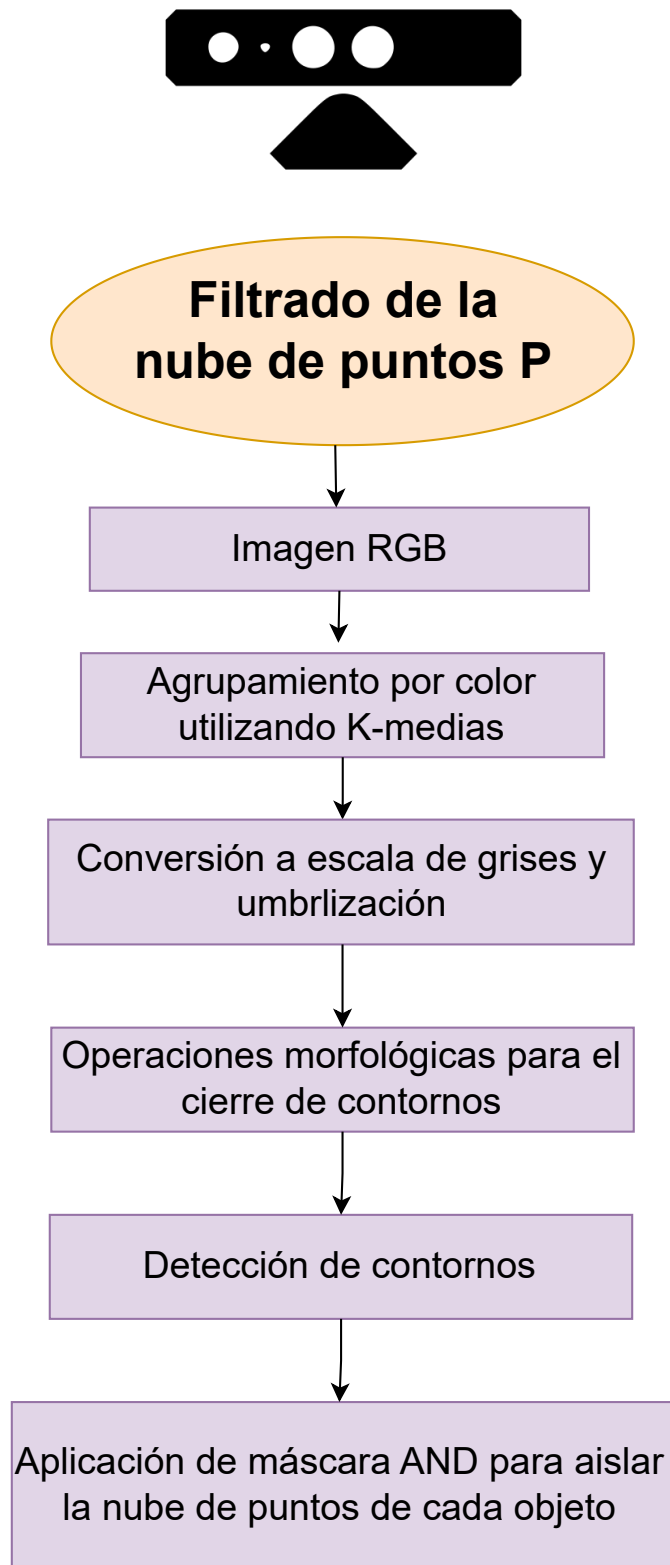


Figura 3.9: Procesamiento de la imagen para la detección de contornos.

humanos suelen tomar objetos, se llega a la conclusión de que es la mayoría de las veces, lo hacen en la dirección perpendicular al eje principal del objeto y cerca de su centro de masa. Ese tipo de agarre es el que se pretende replicar con manipuladores robóticos en este trabajo.



Figura 3.10: Poses naturales de la mano en agarre de objetos.

3.4.3. Procesamiento de la nube de puntos

Sea $P = p_1, p_2, \dots, p_n$ el conjunto de puntos en el espacio cartesiano R^3 que representan la nube de puntos del objeto. Cada punto representa una coordenada (x, y, z) .

Se establece el centroide del conjunto de puntos como el origen del sistema O_{obj} y se calcula con la ecuación

$$p_{centroide} = \frac{1}{n} \sum_{i=1}^n p_i \quad (3.2)$$

Las componentes principales son los valores propios de la matriz de covarianza del conjunto P .

La matriz de covarianza pertenece al grupo de matrices de $n \times n$ sus elementos son las varianzas y las covarianzas de las variables en el espacio R^3 x, y, z .

Cada elemento de la matriz es una es una medida del grado de variación conjunta de dos variables aleatorias(en este caso se tiene tres variables: X, Y, Z), respecto a sus medias. Los elementos que conforman la diagonal principal de la matriz son la varianza de cada variable.

La covarianza expresada en términos de la esperanza matemática $E[X]$ y $E[Y]$ como

$$\text{Cov}(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - E[X]) (y_i - E[Y]) \quad (3.3)$$

Y la matriz de covarianza, los nueve elementos son los valores de covarianza para las coordenadas en R^3 .

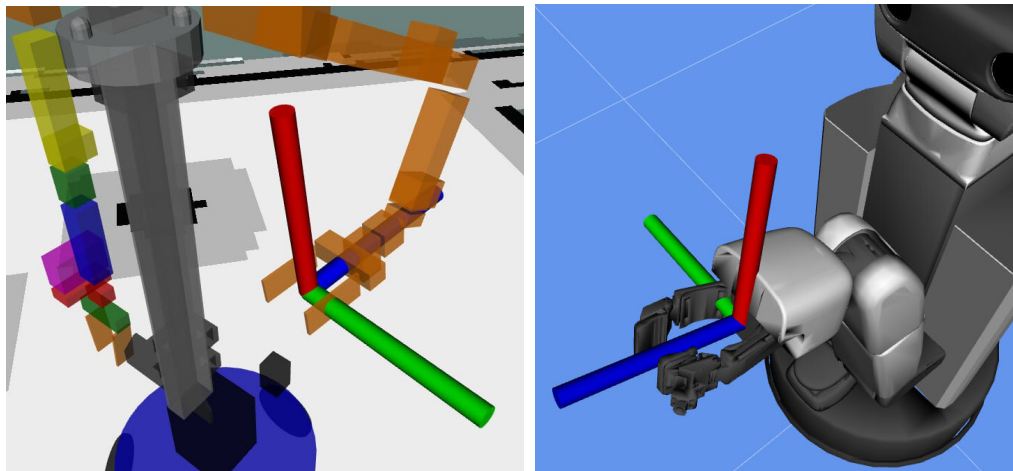
$$C = \begin{pmatrix} \sigma_{XX}^2 & \sigma_{XY}^2 & \sigma_{XZ}^2 \\ \sigma_{YX}^2 & \sigma_{YY}^2 & \sigma_{YZ}^2 \\ \sigma_{ZX}^2 & \sigma_{ZY}^2 & \sigma_{ZZ}^2 \end{pmatrix} \quad (3.4)$$

Los valores propios de la matriz de covarianza $\lambda_1 > \lambda_2 > \lambda_3$ son la primera, segunda y tercera componente principales que brindan información acerca de las direcciones de mayor variación en el conjunto. La interpretación geométrica de estos valores indica que el vector propio asociado al valor propio mas grande, brinda la dirección de la mayor longitud de un objeto.

3.4.4. Construcción del sistema coordenado del objeto

El eje principal y el centroide de la nube de puntos del objeto son utilizados para construir el sistema coordenado asociado. El centroide se utiliza como punto de origen del sistema O_{obj} y el eje principal se asigna al eje x del sistema con el fin de que quede paralelo al eje x del efector final.

Es necesario que a partir de las componentes principales obtenidos con el análisis estadístico, se construya un sistema coordenado adecuado a la configuración del sistema coordenado del efector final del robot, es decir el eje principal del objeto debe coincidir con el eje normal a la pinza o dedos de la mano robótica, tal como se muestra en las figuras ?? y 3.11b.



(a) Sistema asociado al efector final del robot Justina(simulación).
 (b) Sistema asociado al efector final del robot hsrb(simulación).

Figura 3.11: El eje x del sistema coordenado asociado al efector final es perpendicular a las pinzas de agarre.

Para construir el sistema coordenado del objeto se asigna el eigenvector con sentido positivo, que representa a la componente principal, al eje x del sistema coordenado del objeto El eje y y el eje Z del sistema coordenado del objeto se pueden determinar aplicando la ecuación

$$\hat{i}_{obj} = +\hat{E}_1 \quad (3.5)$$

$$\hat{j}_{obj} = \hat{E}_2 \quad (3.6)$$

$$\hat{k}_{obj} = \hat{i}_{obj} \times \hat{j}_{obj} \quad (3.7)$$

Donde

E_1 = eigenvector(positivo) que representa a la primera componente del PCA

E_2 = eigenvector que representa a la segunda componente del PCA

El nuevo sistema coordenado O_{obj} esta referenciado respecto a O_{camara} . Para fines prácticos se cambia el sistema de referencia con el cual esta expresado O_{obj} del sistema de la cámara $O_{realsense}$ al sistema de la base del robot O_{base}

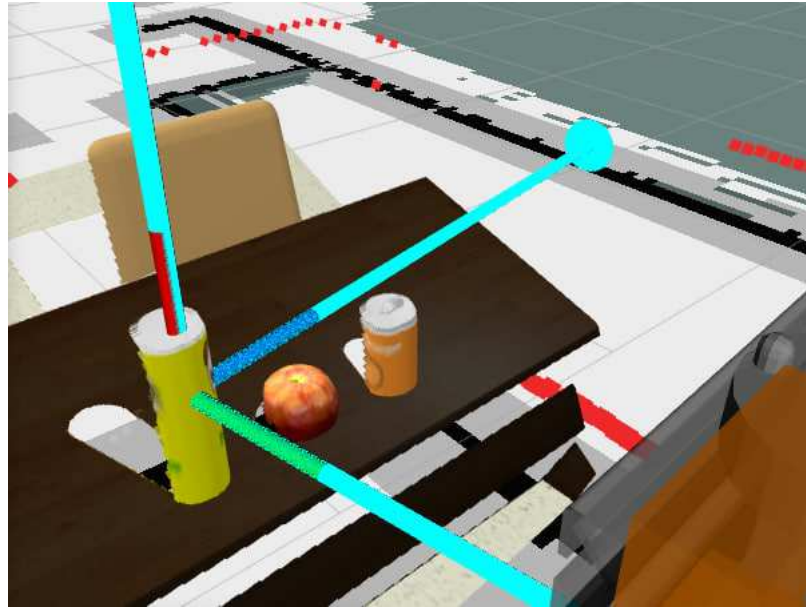


Figura 3.12: Objeto segmentado y su sistema coordenado obtenido a partir del análisis de componentes principales.

3.4.5. Estimación de la geometría del objeto

Al hacer un análisis de las relaciones entre las componentes principales es posible estimar formas geométricas básicas que conforman la caja delimitadora que encierra al conjunto de puntos.

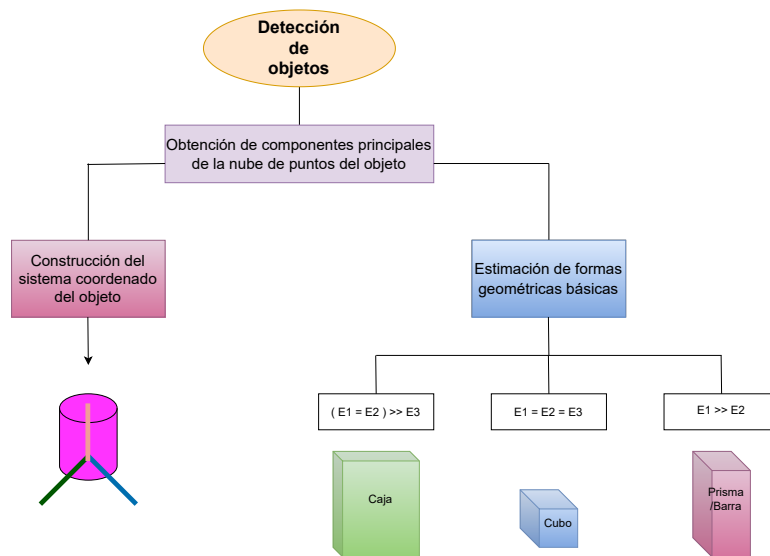


Figura 3.13

Las dimensiones del objeto se estiman a partir de los puntos mínimos y máximos sobre los ejes x_{obj} , y_{obj} y z_{obj}

$$H = x_{max} - x_{min} \quad (3.8)$$

$$W = y_{max} - y_{min} \quad (3.9)$$

$$L = 2 * (z_{max} - z_{min}) \quad (3.10)$$

Donde

H = altura del objeto

W = ancho del objeto

L = profundidad del objeto

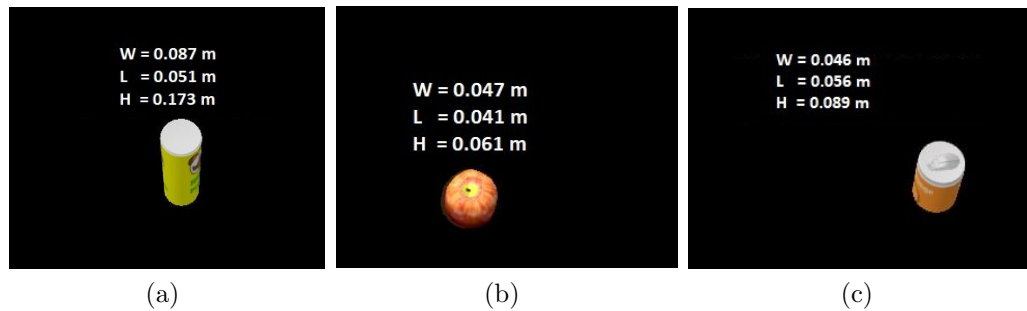


Figura 3.14: Con los valores mínimos y máximos en el eje coordenado del objeto es posible estimar las dimensiones del objeto. Medidas estimadas para pringles 3.14a, medidas estimadas para manzana 3.14b, medidas estimadas para contenedor de jugo 3.14c.

En este capítulo fue descrito el proceso para obtención de información proveniente de la cámara RGB-D para extraer datos de la imagen y de la nube de puntos.

La imagen RGB fue utilizada para abordar el problema de la detección de objetos en escena. Se mostró el proceso seguido para la cuantización por color de la imagen utilizando el algoritmo *k_mmedias*, necesario para facilitar la búsqueda de objetos a partir de sus contornos. También se describió el análisis realizado a la nube de puntos del objeto con PCA para obtener información acerca de las dimensiones de la caja delimitadora del objeto, de su centroide.

Se estableció una clasificación para el tipo de caja delimitadora del objeto en base a sus dimensiones y se obtuvo el centroide del objeto. Por último se explicó el proceso seguido para la creación del sistema coordenado del objeto. Lo cual es de gran importancia pues sirve para generar las herramientas utilizadas mas adelante.

En el siguiente capítulo se utiliza la información ya obtenida del objeto para establecer las reglas de agarre de objetos que forman parte del planeador de movimientos.

Capítulo 4

Planeación de movimientos

El problema de la planeación de movimientos, en forma general consiste en llevar un cuerpo desde una configuración inicial hasta otra configuración final. Existen planeadores de movimientos para la correcta navegación de los robots en entornos humanos, en este trabajo se aborda la planeación de movimientos para un manipulador robótico.

El cerebro humano planifica los movimientos de los brazos en tareas cotidianas como tomar objetos, pues debe determinar como se moverán y el orden en que se debe mover cada articulación antes de tomar un objeto. Una necesidad fundamental en robótica es tener algoritmos que conviertan especificaciones de alto nivel de tareas humanas en descripciones de bajo nivel de cómo moverse. Los términos planificación de movimiento y planificación de trayectoria se utilizan a menudo para este tipo de problemas.

La planificación del movimiento del robot normalmente ignora la dinámica y otras restricciones diferenciales y se centra principalmente en las traslaciones y rotaciones necesarias para mover el objeto[20]. En este capítulo se explica el diseño del sistema de planificación de movimientos.

La primera parte trata sobre la descripción del manipulador robótico utilizado en este trabajo, así como su cinemática directa, la modificación realizada al algoritmo de cinemática inversa existente y el método usado para el diseño de trayectorias del manipulador que conectan al efector final con el objeto deseado. En la segunda parte se explican las reglas de agarre propuestas para objetos desconocidos, teniendo como datos una aproximación de sus dimensiones y el sistema coordinado asociado construido en el capítulo anterior, el capítulo finaliza con el diseño de un controlador proporcional diferencial con compensación de gravedad para el manipulador.

4.1. Descripción del manipulador

Para este trabajo se utilizó un brazo robótico de 7 grados de libertad (GDL), forma parte del robot de servicio Justina, que pertenece al laboratorio de Bio-Robótica de la Facultad de Ingeniería (UNAM). Las articulaciones son de tipo rotacional, y su configuración es antropomórfica.

Esta configuración del manipulador produce una cinemática redundante, la redundancia brinda flexibilidad y una mayor capacidad para evadir obstáculos y así realizar tareas en entornos reducidos de forma similar a los humanos[13].

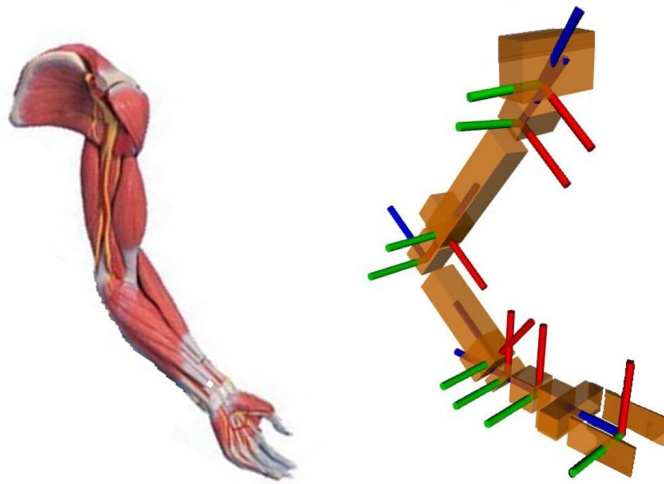


Figura 4.1: A la izquierda, el brazo humano, tiene tres grados de libertad en el hombro, un grado de libertad en el codo y tres grados de libertad en la muñeca. A la derecha, el manipulador robótico del robot Justina, sobre cada articulación se observa un sistema coordinado más el sistema de la herramienta.

4.1.1. Puntos de singularidad

Son aquellas configuraciones del brazo a partir de las cuales ciertas direcciones de movimiento del brazo resultan inalcanzables. En ellas, puede suceder que las velocidades finitas del efector final estén relacionadas con velocidades ilimitadas en el espacio articular o viceversa. En los puntos de singularidad también puede ocurrir que las fuerzas y pares de efector final limitados se correspondan con pares de torsión ilimitados para los actuadores de las articulaciones. Los puntos en el límite del espacio de trabajo del manipulador también se consideran puntos de singularidad. De hecho, siempre

que dos ejes de articulación de revolución en cualquier lugar son colineales, se produce una singularidad, ya que una rotación igual y opuesta alrededor de los ejes no produce movimiento neto del efector final.

Cerca de los puntos de singularidad no existirá una solución única al problema de cinemática inversa. En tales casos puede que no haya solución o que haya infinitas soluciones. Un ejemplo de punto de singularidad puede ser observado en la muñeca esférica. esto sucede cuando los ejes conjuntos z_3 y z_5 son colineales. Esta es la única singularidad de la muñeca esférica y es inevitable sin imponer límites mecánicos al diseño de la muñeca para restringir su movimiento de tal manera que se impida que z_3 y z_5 se alineen.

4.2. Cinemática del manipulador

La cinemática del manipulador se ocupa de responder las dos preguntas básicas siguientes para una geometría de manipulador específica:

1. Para un conjunto dado de configuraciones articulares, ¿cuál es la postura del efector final del manipulador? Esto será respondido por la cinemática directa.
2. ¿Qué ángulos articulares se requieren para alcanzar la postura de efector final deseada? Esto será respondido por la cinemática inversa[2].

4.2.1. Cinemática Directa

El problema de cinemática directa para un manipulador de cadena en serie es encontrar la posición y orientación del efector final X relativa al sistema base, dadas las posiciones de todas las articulaciones (q_1, \dots, q_n) y los valores de todos los parámetros geométricos de los eslabones.

$$X = f(q) \quad (4.1)$$

El problema de la cinemática directa se resuelve calculando la transformación entre un sistema de referencia fijado en el efector final y otro sistema de referencia fijado en la base. La transformación que describe la posición del efector final con respecto a la base se obtiene concatenando transformaciones entre sistemas fijados en eslabones adyacentes de la cadena. Lo cual se reduce a encontrar una matriz de transformación homogénea equivalente de 4×4 que relacione el desplazamiento espacial del sistema de referencia del efector final con el sistema de referencia base[33]. Con frecuencia suele asignarse un sistema fijo en el efector final nombrado sistema de la herramienta, y tiene

un desplazamiento constante tanto en posición como en orientación desde el sistema n. Para el manipulador serial usado en este trabajo el sistema de coordenadas de la herramienta, referido al sistema base se obtiene a partir de la siguiente expresión

$$T_{tool}^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T_7^6 T_{tool}^7$$

T_a^b es la matriz de transformaciones homogéneas conformada por una matriz de rotación $R^{3 \times 3}$ y un vector de posición (px_a^b, py_a^b, pz_a^b) (2.10) que transforma posición y orientación desde el sistema coordenado b al sistema coordenado a.

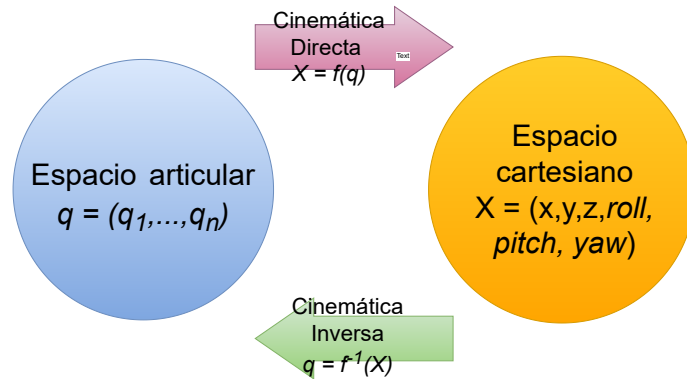


Figura 4.2: Espacio articular y espacio cartesiano.

4.3. Cinemática inversa

El problema de cinemática inversa para un manipulador de cadena en serie es encontrar los valores de las posiciones de las articulaciones dada la posición y orientación del efector final con respecto a la base y dados los valores de todos los parámetros geométricos de los eslabones[33]. La cinemática inversa se describe con la siguiente ecuación

$$q = f^{-1}(X) \quad (4.2)$$

4.4. Cinemática diferencial

El problema de cinemática directa instantánea para un manipulador de cadena en serie es: dadas las posiciones de todos los miembros de la cadena y las velocidades de movimiento alrededor de todas las articulaciones, encuentre la velocidad total del efector final.

La velocidad de movimiento alrededor de la articulación es la velocidad angular de rotación alrededor de una articulación de revolución o la velocidad de traslación de deslizamiento a lo largo de una articulación prismática. La velocidad total de un miembro es la velocidad lineal con respecto al sistema de referencia combinada con su velocidad angular. Es decir, la velocidad total tiene seis componentes independientes y, por lo tanto, representa completamente el campo de velocidades del miembro. Es importante señalar que esta definición incluye el supuesto de que se conoce completamente la posición del mecanismo[3].

4.5. Matriz Jacobiana

La relación entre las velocidades de las articulaciones y la correspondiente velocidad lineal y angular del efector final se describe mediante una matriz, denominada Jacobiano, que depende de la configuración del manipulador[23].

El jacobiano constituye una de las herramientas más importantes para la caracterización del manipulador; es útil para encontrar singularidades, analizar redundancia, determinar algoritmos de cinemática inversa, describir el mapeo entre las fuerzas aplicadas al efector final y los pares resultantes en las articulaciones (estática) y, para derivar ecuaciones dinámicas de movimiento.

4.5.1. Jacobiano Geométrico

Se desea expresar la velocidad lineal del efector final \dot{p} y la velocidad angular ω en función de las velocidades de articulación \dot{q}

$$p_{ef} = J_p(q)\dot{q} \quad (4.3)$$

$$\omega_{ef} = J_O(q)\dot{q} \quad (4.4)$$

Donde $J_p \in \mathbb{R}^{3 \times n}$ relaciona la contribución de las velocidades articulares con la velocidad lineal del efector final p_{ef} , mientras que J_O es la matriz de $3 \times n$ que relaciona la contribución de las velocidades articulares con la velocidad angular del efector final ω_{ef} . En forma compacta se puede escribir como

$$\dot{X} = \begin{pmatrix} \dot{p}_{ef} \\ \dot{\omega}_{ef} \end{pmatrix} = J(q)\dot{q} \quad (4.5)$$

La cual representa la ecuación cinemática diferencial del manipulador. J es una matriz de $6 \times n$.

$$J = \begin{pmatrix} J_P \\ J_O \end{pmatrix} \quad (4.6)$$

- Si $n > m$: Se trata del Jacobiano de un robot redundante.
- Si $n < m$: El robot tiene menos articulaciones n que coordenadas del espacio de trabajo, o ha caído en punto de singularidad.

4.5.2. Jacobiano Analítico

Alternativamente, si la configuración espacial del efector final se expresa con referencia a una representación mínima en el espacio de trabajo, es posible calcular la matriz jacobiana mediante la diferenciación de la función cinemática directa con respecto a las variables articulares. El jacobiano resultante, denominado jacobiano analítico, en general difiere del geométrico.

La velocidad de traslación del sistema coordinado del efector final se puede expresar como la derivada en el tiempo del vector p_{ef} , que representa el origen del sistema del efector final con respecto al sistema base base, es decir

$$\dot{p}_{ef} = \frac{\partial p_{ef}}{\partial q} \dot{q} = J_p(q)\dot{q} \quad (4.7)$$

En cuanto a la velocidad de rotación del sistema del efector final, se puede considerar la representación mínima de la orientación en términos de tres variables ϕ_{ef} .

$$\dot{\phi}_{ef} = \frac{\partial \phi_{ef}}{\partial q} \dot{q} = J_\phi(q)\dot{q} \quad (4.8)$$

Su derivada temporal $\dot{\phi}_{ef}$ en general difiere del vector de velocidad angular definido anteriormente.

El cálculo del jacobiano $J_\phi(q)$ como $\frac{\partial \phi_{ef}}{\partial q}$ no es sencillo, ya que la función $\phi_{ef}(q)$ generalmente no está disponible en forma directa, requiere el cálculo de los elementos de la matriz de rotación relativa. Sobre estas premisas, la ecuación de cinemática diferencial se puede obtener como la derivada en el tiempo de la ecuación de cinemática directa, es decir

$$\dot{X} = \begin{pmatrix} \dot{p}_{ef} \\ \dot{\phi}_{ef} \end{pmatrix} = \begin{pmatrix} J_p(q) \\ J_\phi(q) \end{pmatrix} \dot{q} = J_A(q) \dot{q} \quad (4.9)$$

Donde el jacobiano analítico

$$J_A(q) = \frac{\partial f(q)}{\partial q} \quad (4.10)$$

Es diferente del jacobiano geométrico J , ya que la velocidad angular del efector final ω_{ef} con respecto al sistema base está dada por

$$\omega_{ef} = T(\phi_{ef}) \dot{\phi}_{ef} \quad (4.11)$$

Los ángulos de Euler pueden ser usados para expresar la orientación ϕ_{ef} .

Respecto al uso de uno u otro en todos aquellos problemas donde la influencia del jacobiano importa, se usa el jacobiano geométrico siempre que sea necesario hacer referencia a cantidades de significado físico claro, mientras que se usa el jacobiano analítico siempre que sea necesario hacer referencia a cantidades diferenciales de variables definidas en el espacio de trabajo[23].

4.6. Algoritmo de cinemática inversa

Partiendo de la ecuación de cinemática directa 4.1, asumiendo que es diferenciable y X_d es el vector de coordenadas deseadas del efector final. Para el método de N-R se formula la ecuación

$$g(q) = X_d - f(q) = 0 \quad (4.12)$$

El objetivo es encontrar las coordenadas q_d , tal que

$$g(q_d) = X_d - f(q_d) = 0 \quad (4.13)$$

Dada una suposición inicial q^0 la cual es cercana a la solución q_d , si se desarrolla la cinemática directa como una expansión de la serie de Taylor

$$X_d = f(q_d) = f(q^0) + \frac{\partial f}{\partial q} \Big|_{q^0} (q_d - q^0) + \dots \quad (4.14)$$

Que puede re-expresarse como

$$X_d = f(q_d) = f(q^0) + J(q^0) \Delta q + \dots \quad (4.15)$$

Donde $J(q^0)$ es el jacobiano evaluado en q^0 .

Tomando los dos primeros términos de la expansión en series de Taylor, es posible aproximar la ecuación 4.15

$$X_d = f(q^0) + J(q^0)\Delta q \quad (4.16)$$

Reordenando

$$J(q^0)\Delta q = X_d - f(q^0) \quad (4.17)$$

Asumiendo que el Jacobiano es invertible, se puede resolver la ecuación para Δq

$$\Delta q = J^{-1}(q^0)(X_d - f(q^0)) \quad (4.18)$$

4.6.1. Matriz Jacobiana inversa

Cuando $m \neq n$, no existirá la inversa del Jacobiano porque no será cuadrada, en su lugar puede utilizarse la matriz pseudo inversa del Jacobiano J^\dagger .

La matriz seudo inversa es aquella que satisface las condiciones de Moore Penrose[17].

A partir de la matriz transpuesta

$$J_A J_A^\dagger J_A = J_A$$

$$J_A^\dagger J_A J_A^\dagger = J_A^\dagger$$

$$(J_A J_A^\dagger)^T = J_A J_A^\dagger$$

$$(J_A^\dagger J_A)^T = J_A^\dagger J_A$$

Si $m < n$ J_A es rectangular y de rango completo, su pseudo inversa se puede calcular como

$$J_A^\dagger = J_A^T (J_A J_A^T)^{-1} \quad (4.19)$$

Si J_A es cuadrada, la expresión 4.19 se reduce a la matriz inversa estándar.

4.6.2. Soluciones cerradas

Las soluciones de forma cerrada cuentan con la ventaja de ser más rápidas que las soluciones numéricas, a su vez tienen la desventaja de no ser una solución general ya que depende en gran medida de la geometría del robot. Ejemplos de soluciones cerradas son el enfoque geométrico, adecuado para robots con pocos grados de libertad, consiste en encontrar los valores articulares q analizando la geometría del robot. En el enfoque algebraico consiste en encontrar los valores articulares a partir de las ecuaciones de cinemática directa y el desacople cinemático, adecuado para robots de 6GDL y muñeca esférica, se busca obtener la posición con las primeras tres articulaciones y la orientación con las últimas tres.

4.6.3. Soluciones numéricas

Los métodos numéricos no dependen del robot, por lo que pueden aplicarse a cualquier estructura cinemática. Las desventajas de los métodos numéricos consisten en que pueden ser más lentos para encontrar una solución, y en algunos casos, no permiten calcular todas las soluciones posibles. Los más comunes se pueden dividir en categorías de métodos de eliminación simbólica, métodos de continuación y métodos iterativos.

4.6.4. El método de Newton-Raphson

La naturaleza de los métodos iterativos los vuelve adecuados para ser programados en computadora. Uno de los más utilizados es el método de Newton-Raphson, también conocido como método de las tangentes. Funciona con aproximaciones sucesivas, pues a partir de una primera aproximación a la solución de una ecuación y aplicando en forma repetida una fórmula de recurrencia, obtiene una mejor aproximación a la solución[1].

Si el valor inicial para la raíz es x_i , entonces se puede trazar una tangente desde el punto

$$[x_i, f(x_i)]$$

de la curva. Por lo común, el punto donde esta tangente cruza al eje x x_{i+1} representa una aproximación mejorada de la raíz. De la primera derivada[5].

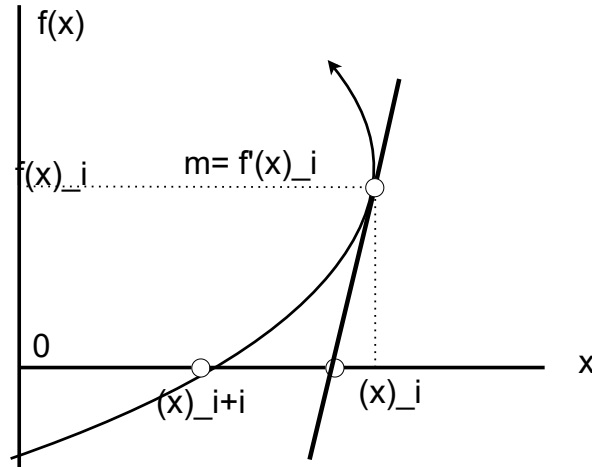


Figura 4.3: La primera derivada en x es equivalente a la pendiente.

$$f'(x_i) = -\frac{f(x_i) - 0}{x_i - x_{i+1}} \quad (4.20)$$

Se obtiene la siguiente aproximación

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_{i+1})} \quad (4.21)$$

la cual se conoce como fórmula de Newton-Raphson.

4.6.5. Cálculo numérico del Jacobiano

Si la función de cinemática directa $X = f(q)$ mapea del espacio articular al espacio de trabajo $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$, la matriz jacobiana J_A será una matriz $J_A \in \mathbb{R}^{m \times n}$.

$$J_A(x) = \begin{pmatrix} \frac{\partial X(1)}{\partial q_1} & \cdots & \frac{\partial X(1)}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial X(m)}{\partial q_1} & \cdots & \frac{\partial X(m)}{\partial q_n} \end{pmatrix} \quad (4.22)$$

Donde X = vector de configuración en el espacio de trabajo $X = [(x, y, z, \psi, \theta, \phi)]$, con $m = 1, \dots, 6$.

Para el manipulador de 7GDL ($n > m$), la matriz jacobiana no es cuadrada, sino una matriz $J \in \mathbb{R}^{6 \times 7}$.

$$J_A(x) = \begin{pmatrix} \frac{\partial x}{\partial q_1} & \frac{\partial x}{\partial q_2} & \frac{\partial x}{\partial q_3} & \frac{\partial x}{\partial q_4} & \frac{\partial x}{\partial q_5} & \frac{\partial x}{\partial q_6} & \frac{\partial x}{\partial q_7} \\ \frac{\partial y}{\partial q_1} & \frac{\partial y}{\partial q_2} & \frac{\partial y}{\partial q_3} & \frac{\partial y}{\partial q_4} & \frac{\partial y}{\partial q_5} & \frac{\partial y}{\partial q_6} & \frac{\partial y}{\partial q_7} \\ \frac{\partial z}{\partial q_1} & \frac{\partial z}{\partial q_2} & \frac{\partial z}{\partial q_3} & \frac{\partial z}{\partial q_4} & \frac{\partial z}{\partial q_5} & \frac{\partial z}{\partial q_6} & \frac{\partial z}{\partial q_7} \\ \frac{\partial \psi}{\partial q_1} & \frac{\partial \psi}{\partial q_2} & \frac{\partial \psi}{\partial q_3} & \frac{\partial \psi}{\partial q_4} & \frac{\partial \psi}{\partial q_5} & \frac{\partial \psi}{\partial q_6} & \frac{\partial \psi}{\partial q_7} \\ \frac{\partial q_1}{\partial \theta} & \frac{\partial q_2}{\partial \theta} & \frac{\partial q_3}{\partial \theta} & \frac{\partial q_4}{\partial \theta} & \frac{\partial q_5}{\partial \theta} & \frac{\partial q_6}{\partial \theta} & \frac{\partial q_7}{\partial \theta} \\ \frac{\partial q_1}{\partial \phi} & \frac{\partial q_2}{\partial \phi} & \frac{\partial q_3}{\partial \phi} & \frac{\partial q_4}{\partial \phi} & \frac{\partial q_5}{\partial \phi} & \frac{\partial q_6}{\partial \phi} & \frac{\partial q_7}{\partial \phi} \\ \frac{\partial q_1}{\partial \theta} & \frac{\partial q_2}{\partial \theta} & \frac{\partial q_3}{\partial \theta} & \frac{\partial q_4}{\partial \theta} & \frac{\partial q_5}{\partial \theta} & \frac{\partial q_6}{\partial \theta} & \frac{\partial q_7}{\partial \theta} \end{pmatrix} \quad (4.23)$$

Cada columna representa el cambio de una variable del espacio de trabajo con respecto a una misma articulación.

Es posible aproximar el jacobiano con diferencias finitas si a Δq se le asigna un valor suficientemente pequeño para realizar una buena aproximación de la derivada. La aproximación para cada elemento de la matriz 4.23 es

$$\frac{\partial X}{\partial q_i} \approx \frac{\Delta X}{\Delta q_i} = \frac{\Delta f(q)}{\Delta q_i} = \frac{f(q_1, \dots, q_i + \Delta q, \dots, q_n) - f(q_1, \dots, q_i - \Delta q, \dots, q_n)}{2\Delta q} \quad (4.24)$$

Donde q_i es la articulación con respecto a la cual esta cambiando una variable del espacio de trabajo.

Aunque en general el método de Newton-Raphson es muy eficiente, hay situaciones donde se comporta de manera deficiente. Por ejemplo, si la estimación inicial se encuentra lejos de la solución, el método puede no converger e iterar de forma infinita. De manera que no hay un criterio general de convergencia para el método de Newton-Raphson. Su convergencia depende de la naturaleza de la función y de la exactitud del valor inicial. La única solución en estos casos es tener un valor inicial que sea “suficientemente” cercano a la raíz[4].

4.6.6. Planeación de trayectorias para mejorar la convergencia de N-R

Como ya se explicó, para que el método pueda converger a una solución es importante que la estimación inicial se encuentre cerca de la solución. Esto implica conocer de manera aproximada los valores del vector de posiciones q para la solución que se desea conocer dado una configuración en el espacio cartesiano $(x, y, z, roll, pitch, yaw)$, en consecuencia, al aplicar el método para una posición y orientación alejada de la posición inicial, iterará infinito número de veces, ya que solo proporciona la solución para posturas cercanas a la inicial, es posible generar una serie de puntos en el espacio cartesiano, separados una mínima distancia y que conecten un vector de posiciones iniciales $(x_i, y_i, z_i, roll, pitch, yaw)$, con cualquier vector de posiciones finales

$(x_f, y_f, z_f, roll, pitch, yaw)$ dentro del espacio de trabajo. De esta manera es posible obtener la solución de cinemática inversa para cualquier punto del espacio de trabajo, siempre y cuando exista la matriz jacobiana. Donde la solución para el primer punto es obtenida a partir de las lecturas de los encoders de los actuadores del brazo, para los siguientes puntos se calcula la cinemática inversa, usando el punto previo como condición inicial. De este modo la condición inicial siempre está cerca de la solución buscada.

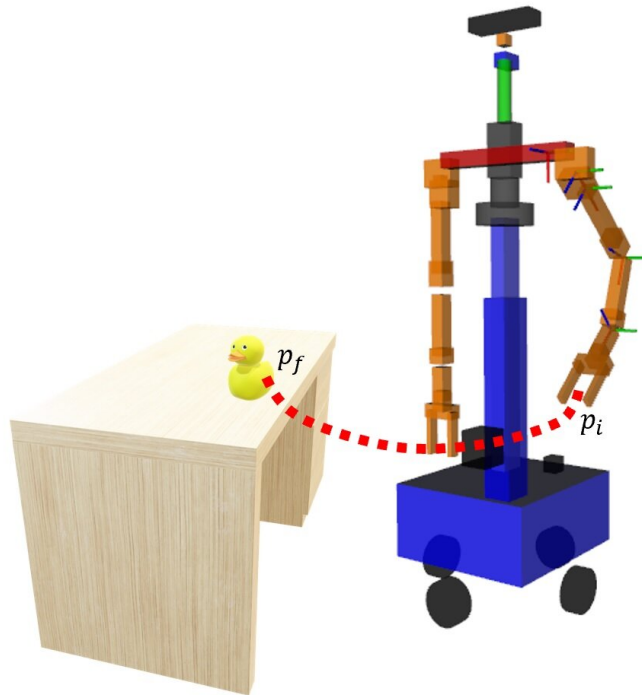


Figura 4.4: La cinemática inversa puede ser obtenida para un punto distante p_f si primero se obtiene para una serie de puntos generados con el fin de conectarlo con el punto inicial p_i de cual se tiene una buena estimación q .

4.7. Planeación de trayectorias

Las trayectorias suelen diseñarse en el espacio cartesiano ya que al ser humano le resulta familiar, a diferencia del espacio articular, por otro lado, el espacio articular resulta idóneo para el desarrollo de un control de posición, pues los sensores(encoders) que registran la posición de los actuadores entregan valores en radianes.

El trabajo del planeador de trayectorias consiste en especificar la secuencia de configuraciones correspondientes al efector final, desde una configuración inicial hasta una configuración final dentro del espacio de trabajo (W), satisfaciendo condiciones iniciales y finales de velocidad y aceleración en los actuadores. Genera una trayectoria con elementos finitos, en cada uno de los cuales esta especificada una configuración del efector final. El planeador se diseña como un sistema cuyas entradas son

- Tiempo de la trayectoria $t_i - t_f$.
- Numero de elementos de la trayectoria n .
- Configuración inicial $(x_i, y_i, z_i, roll_i, pitch_i, yaw_i)$.
- Configuración final $(x_f, y_f, z_f, roll_f, pitch_f, yaw_f)$.
- Condiciones iniciales de posición, velocidad y aceleración $p(t_i), v(t_i), \alpha(t_i)$.
- Condiciones finales de posición, velocidad y aceleración $p(t_f), v(t_f), \alpha(t_f)$.

Con los valores de entrada el sistema se encarga de generar un perfil de velocidades y aceleraciones, a través de funciones polinomiales de grado m . Estos polinomios son parametrizados en el tiempo $q^m(t)$, el grado del polinomio viene dado por el número de restricciones m en posición, velocidad y aceleración impuestos en el estado inicial

$$p(t_i) = p_0 \quad (4.25)$$

$$v(t_i) = v_0 \quad (4.26)$$

$$\alpha(t_i) = \alpha_0 \quad (4.27)$$

Y en el estado final

$$p(t_f) = p_f \quad (4.28)$$

$$v(t_f) = v_f \quad (4.29)$$

$$\alpha(t_f) = \alpha_f \quad (4.30)$$

Se tienen seis restricciones, un polinomio que satisface las restricciones anteriores contiene seis términos independientes y esta dado por la expresión

$$p(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \quad (4.31)$$

las funciones correspondientes a la velocidad y aceleración se obtienen derivando $p(t)$

$$v(t) = a_1 + 2 * a_2 t + 3 * a_3 t^2 + 4 * a_4 t^3 + 5 * a_5 t^4 \quad (4.32)$$

$$\alpha(t) = 2 * a_2 + 6 * a_3 t + 12 * a_4 t^2 + 20 * a_5 t^3 \quad (4.33)$$

Un polinomio de quinto orden garantiza que la función de posición tenga segundas derivadas y se obtengan curvas suaves. de ser menor el grado del polinomio las discontinuidades en la aceleración provocarían sacudidas impulsivas, que resultarían en perturbaciones en el sistema y reducirían la precisión de seguimiento. Por esta razón, se contemplan restricciones sobre la aceleración, así como sobre la posición y la velocidad.

Se forma el sistema de ecuaciones

$$\begin{aligned} p_i &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ p_f &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ v_i &= a_1 + 2 * a_2 t + 3 * a_3 t^2 + 4 * a_4 t^3 + 5 * a_5 t^4 \\ v_f &= a_1 + 2 * a_2 t + 3 * a_3 t^2 + 4 * a_4 t^3 + 5 * a_5 t^4 \\ \alpha_i &= 2 * a_2 + 6 * a_3 t + 12 * a_4 t^2 + 20 * a_5 t^3 \\ \alpha_f &= 2 * a_2 + 6 * a_3 t + 12 * a_4 t^2 + 20 * a_5 t^3 \end{aligned}$$

expresado en forma matricial

$$MX = B \quad (4.34)$$

$$\begin{pmatrix} 1 & t_i & t_i^2 & t_i^3 & t_i^4 & t_i^5 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_i & 3t_i^2 & 4t_i^3 & 5t_i^4 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_i & 12t_i^2 & 20t_i^3 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} = \begin{pmatrix} q_i \\ q_f \\ v_i \\ v_f \\ \alpha_i \\ \alpha_f \end{pmatrix} \quad (4.35)$$

puede resolverse con la ecuación 4.36 siempre que exista la matriz inversa[9]

$$X = M^{-1}B \quad (4.36)$$

Cada punto de la trayectoria $(x, y, z, roll, pitch, yaw)$ pasa a la función de cinemática inversa, que entrega puntos con coordenadas en el espacio articular (q_1, \dots, q_n) .

El sistema entrega a la salida una trayectoria para cada articulación del manipulador, en posición, velocidad y aceleración muestreadas para una una cantidad de n muestras especificadas a la entrada. ***Incluir perfiles

4.8. Reglas de agarre propuestas

4.8.1. Utilización de la caja delimitadora del objeto

Con la clasificación realizada a partir de la información del tipo de caja delimitadora del objeto obtenida en el capítulo anterior, es posible definir un conjunto de reglas de agarre para cada tipo de caja.

- Objeto planos o con poco grosor (bidimensionales): La postura de agarre se encuentra por completo definida, se realiza un agarre lateral si la caja es mas alta que ancha y un agarre superior si la caja es mas ancha que alta.
- Objeto cilíndrico/barra: En este caso, existen una gran cantidad de posturas de agarre y es necesario definir una regla para la construcción de candidatos de agarre.
- Objeto esférico/ cuboide: Es posible realizar un agarre lateral o superior para tomar el objeto, pero el agarre superior es el mas acorde a la forma en que las personas toman objetos pequeños, además de que disminuye las probabilidades de choque de la pinza del manipulador con el borde de la mesa al querer tomar el objeto, a diferencia del agarre lateral.

4.8.2. Utilización de la orientación del objeto

La información de la orientación del objeto es utilizada para dos cosas:

- Conocer la inclinación del objeto: Se puede calcular el ángulo de inclinación de la componente principal del objeto y así clasificar en uno de dos estados: vertical u horizontal, los cuales serán utilizados en las reglas de agarre.
- Generar el conjunto de configuraciones candidatas de agarre de la pinza.

4.8.3. Inclinación del objeto

La inclinación del objeto se conoce a partir del cálculo del ángulo entre el eje x del objeto X_{obj} (paralelo a la primera componente principal) y el plano π que representa a la superficie, ambos referidos al sistema base O_b .

Sea el vector n normal al plano π definido por $(0, 0, 1)$, y un vector u colineal a X_{obj} , el ángulo de inclinación θ esta dado por

$$\theta = \arcsin\left(\frac{\vec{n} \cdot \vec{u}}{|\vec{n}| |\vec{u}|}\right) \quad (4.37)$$

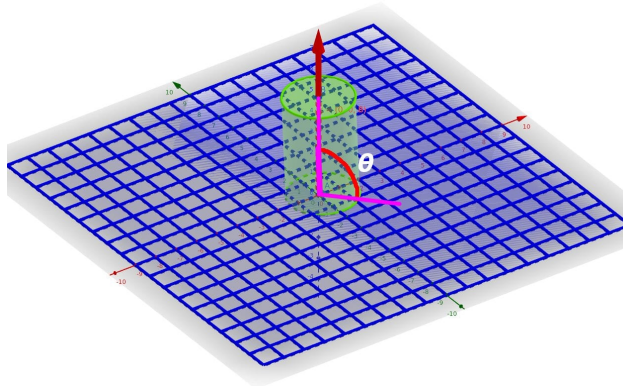


Figura 4.5: Ángulo θ entre el eje x del objeto y la superficie.

Se establece un rango de valores en el ángulo de inclinación, dentro del cual se considera al objeto en estado horizontal, fuera del mismo, se considera al objeto en estado vertical.

Si se considera que cuando $\theta = 1,57 \text{ rad}$ el objeto se encuentra vertical sobre la superficie, se puede establecer que, para que el objeto se considere horizontal con respecto a la superficie $[0 < \theta < 0,532] \text{ rad}$

4.8.4. Utilización del sistema coordenado del objeto

El sistema coordenado del objeto es utilizado como guía para la generación del conjunto de sistemas coordenados candidatos del efector final. El sistema del efector final se alineará con el sistema coordenado del candidato elegido por el planeador de movimientos.

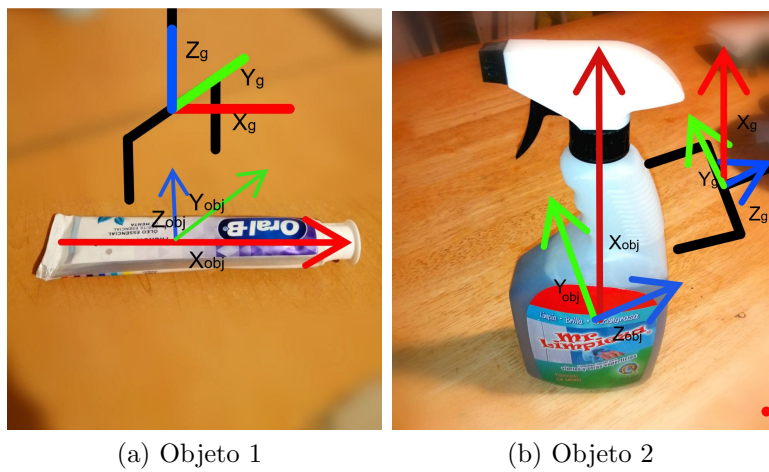


Figura 4.6: Orientación deseada del manipulador en función de la orientación del objeto. La flecha roja sobre los objetos indica la dirección de la dimensión mayor (alineada con el eje X_{obj}). Se busca que los sistemas coordenados del objeto y del efector final estén alineados, la pinza del efector final sea alineada a lo largo de la dirección perpendicular al mayor eje del objeto 4.6.

4.8.5. Tipos de agarre definidos

Con la información de la inclinación del objeto se establecen dos tipos de agarre de objetos: si la inclinación es horizontal, se propone un agarre superior. Si la inclinación del objeto es vertical se propone un agarre lateral 4.7.

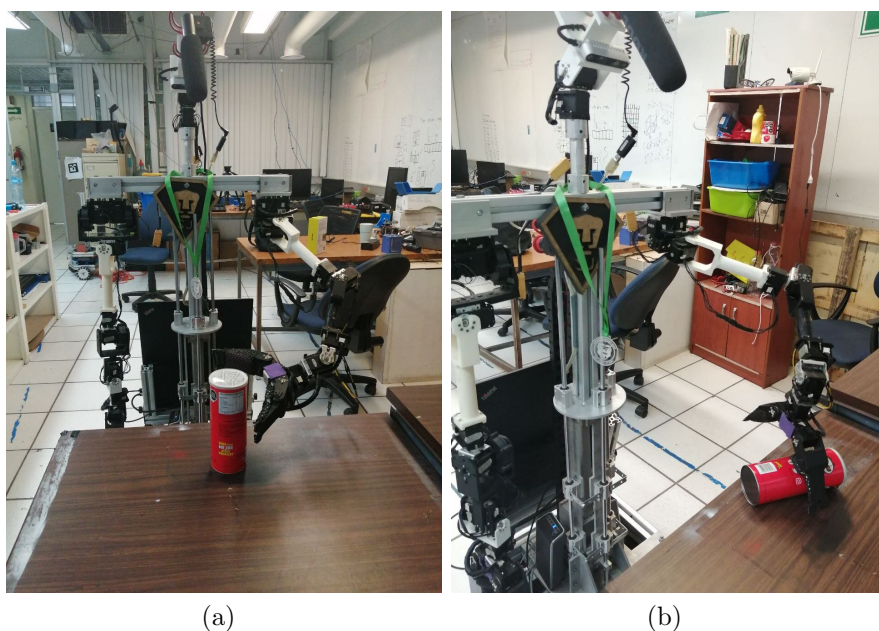


Figura 4.7: Robot Justina realizando agarre lateral y superior.

4.8.6. Implementación de las reglas de agarre

Para objetos con caja delimitadora de tipo prismáticos/barra y cúbicos/esferoides, se asume que la anchura y profundidad (segunda y tercera componente del PCA) es menor a la máxima distancia de apertura del efector final). Se establecen las siguientes tres reglas de agarre.

4.8.7. Regla de agarre para objetos tipo cilindro/barra

Con este tipo de objetos no existe un único sistema coordinado obtenido a partir del PCA, si no, todo un espacio de poses de poses, es decir, el efector final puede acercarse al objeto en una gran cantidad de formas, de las cuales se delimitan los candidatos de agarre a un subespacio de poses.

En términos geométricos este gran conjunto de poses se explica porque en el momento de la construcción del sistema del objeto, existe un único vector que define al eje x del objeto X_{obj} pero existe una infinidad de vectores que pueden asignarse como eje Y_{obj} y eje Z_{obj} del objeto.

En esta sección se propone un método para la generación de sistemas coordinados candidatos de agarre para el sistema del efector final. El espacio de configuraciones de agarre es muy alto, para mantener ese número bajo en

favor de un agarre rápido, se limitan los candidatos a candidatos de agarre a aquellos con origen a la altura del centroide del objeto.

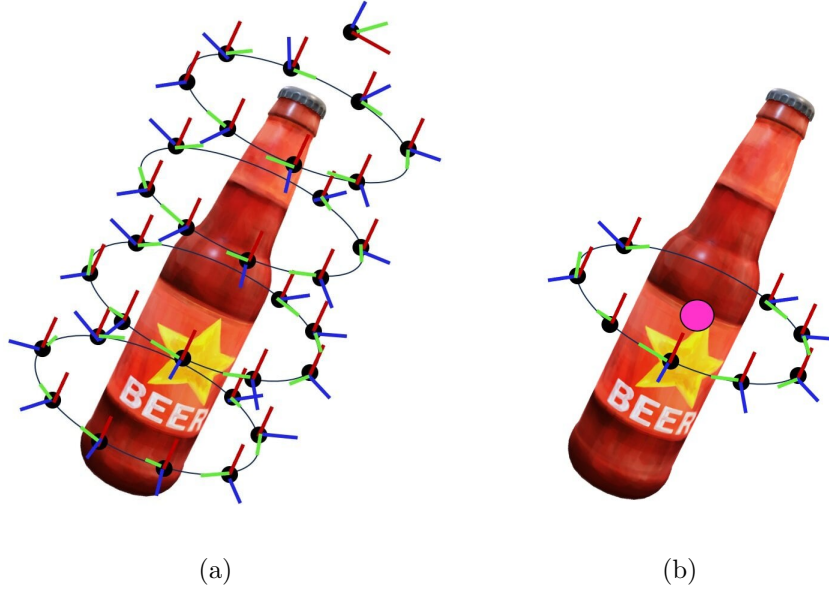


Figura 4.8: (a)Espacio de agarres. (b)Subespacio de agarres al nivel del centroide.

Generación de los candidatos de agarre

Dada una nube de puntos de vista parcial del objeto detectado \mathbb{C} , expresada en el sistema coordenado del objeto. Sea \hat{v} el eje principal del objeto, el centroide del objeto c y H, L, W , las dimensiones de altura, anchura y profundidad de la caja delimitadora del objeto(BB).

Se propone la generación de puntos de agarre dentro del subespacio $\mathbb{D} \subset \mathbb{C}$, el cual resulta de la intersección de una esfera de radio r y un plano de corte π . En el sistema coordenado del objeto las coordenadas del centroide son $p_c = (0, 0, 0)$ La ecuación cartesiana de la esfera con centro en el origen(4.9a) queda determinada por

$$x^2 + y^2 + z^2 = r^2 \quad (4.38)$$

El plano π esta determinado por el punto $p_c = (x_c, y_c, z_c)$ que representa al centroide, y el vector normal \widehat{n}_π , paralelo a \hat{v}

$$x = 0 \quad (4.39)$$

La región de intersección ??

$$z = r \cos(\theta) \quad (4.40)$$

$$y = r \sin(\theta) \quad (4.41)$$

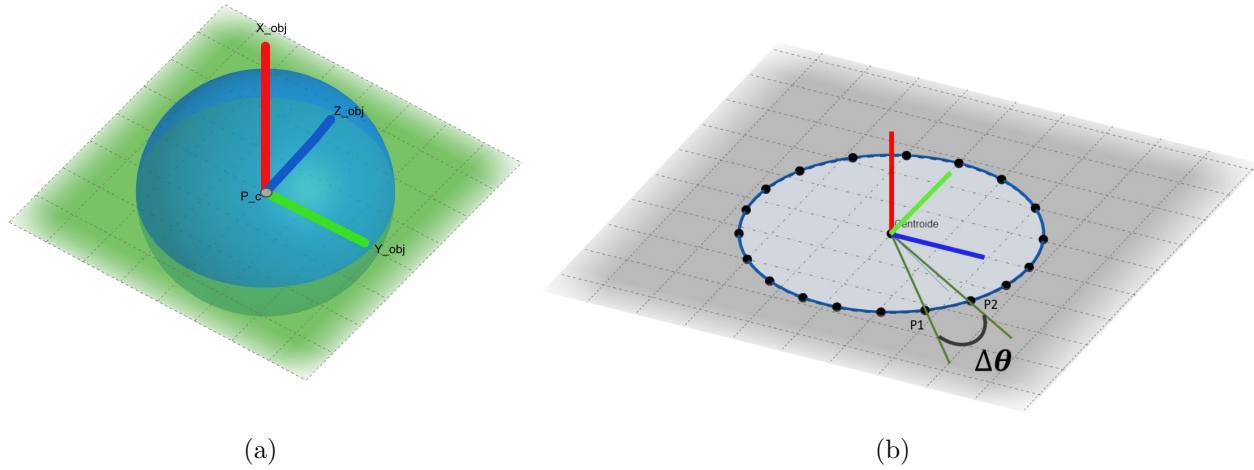


Figura 4.9: (a) Esfera y plano π . (b) Puntos de agarre en la región de intersección de la esfera y el plano.

Donde

$$r = 0.5 L$$

L = Representa la anchura del objeto.

El número de puntos de agarre queda determinado por el tamaño de paso $\Delta\theta$. El i -ésimo punto de agarre se genera con la ecuación

$$P_i = (0, r * \sin(\Delta\theta i), r * \cos(\Delta\theta i)) \quad (4.42)$$

Sobre cada punto de agarre se construye un sistema coordenado que será una posible configuración de agarre para el efector final.

Sea el i -ésimo punto (x_i, y_i, z_i)

1. Se construye el eje x de candidato i -ésimo paralelo al eje x del objeto x_{obj} y con origen en el punto i -ésimo.
2. Eje z : es el vector que va desde el origen al punto n .
3. Eje y = es el vector generado por el producto vectorial entre eje x y eje z del objeto.
4. La cinemática inversa prueba la validez de los candidatos de agarre.

5. Conforme a un criterio de rapidez se elije el primer candidato de agarre aprobado por la cinemática inversa.

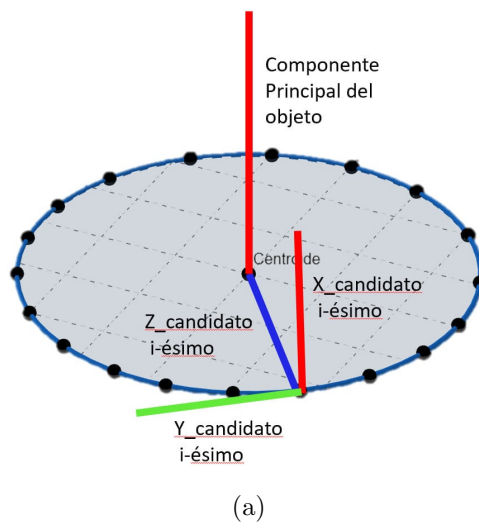


Figura 4.10: Construcción del i -ésimo sistema coordenado candidato para el efector final.

Cuando la mayor dimensión del objeto se encuentra a lo largo de la superficie (objeto volcado), se define un tipo de agarre superior y los candidatos de agarre se generan con la siguiente regla:

Sea el punto de agarre $P_g = (x_g, y_g, z_g)$ el origen de todos los candidatos de agarre, el candidato i -ésimo se genera por la rotación en pitch del sistema del objeto un tamaño de paso $\Delta\theta_i$, con $i = 1, 2, 3, \dots, n$.

Conforme a un criterio de rapidez se elije el primer candidato de agarre aprobado por la cinemática inversa.

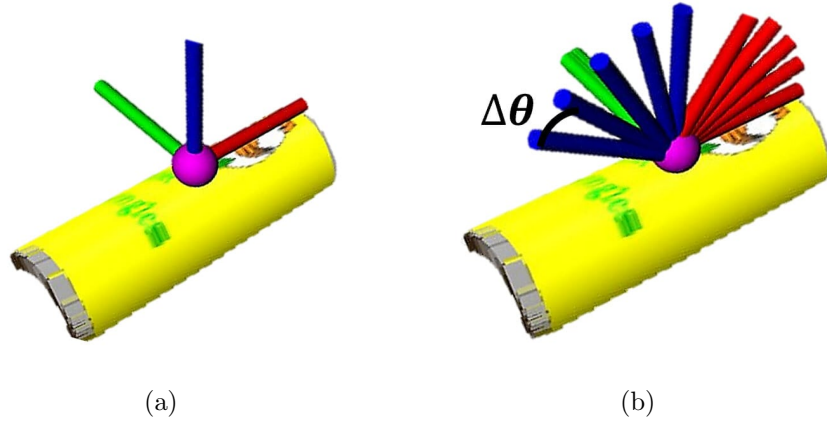


Figura 4.11: (a) Sistema ccordenado del objeto O_{obj} . (b) Generación de candidatos a partir del sistema del objeto $c_1, c_2, c_3, \dots, c_n$.

4.8.8. Objeto plano/caja

Cuando el robot debe manipular objetos planos la orientación del objeto se encuentra completamente definida. En este caso es posible tanto el agarre superior como el lateral. Los posibles punto de agarre se encuentran ahora en los bordes del objeto. En este caso los candidatos se generan no por la gran cantidad de formas de tomar el objeto, sino, por la dificultad que presente la configuración de agarre en la orientación del objeto, y en consecuencia la existencia de una solución de cinemática inversa.

Esta situación se representa en la mano humana al agarrar objetos, es decir, si el objeto se encuentra cercano la postura de la mano sigue la regla de tomarlo de manera que los dedos queden perpendiculares a la mayor dimensión del objeto, pero a medida que incrementa la distancia entre el objeto y la mano, su postura se va perdiendo con el fin de mantener la capacidad de tomar el objeto.

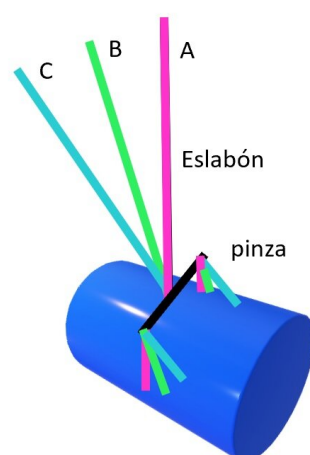


Figura 4.12: Configuraciones del efector final que incrementan las probabilidades de un agarre exitoso.

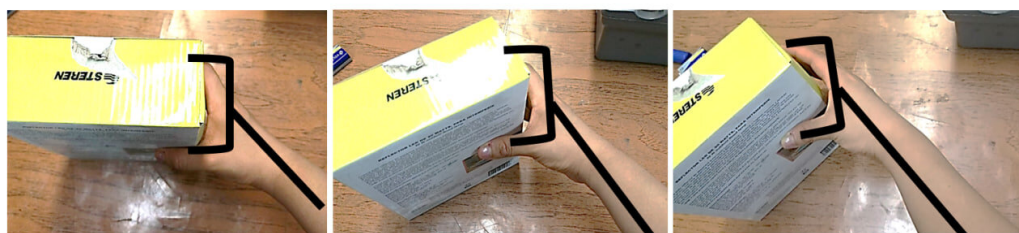


Figura 4.13: A medida que la orientación de la caja cambia, se incrementa la dificultad para alinear el sistema del efector final con su sistema coordinado, por lo que se proponen otras configuraciones para incrementar las probabilidades de agarres exitosos.

Para tener mayores probabilidades de un agarre exitoso se propone un conjunto de agarres candidatos basados en la postura de la mano humana, imitando esas posturas con el efector final, se genera un conjunto de agarres candidatos realizando una rotación en Pitch para el agarre superior, y una rotación en Raw para el agarre lateral.

Punto de agarre

Dada una nube de puntos de vista parcial del objeto detectado \mathbb{C} , expresada en el sistema coordinado del objeto. Sea \hat{v} el eje principal del objeto, el

centroide del objeto c y H, L, W , las dimensiones de altura, anchura y profundidad de la caja delimitadora del objeto (BB). Se propone construir un conjunto de configuraciones candidatas asociados al efector final, cada elemento de este conjunto será un candidato de agarre evaluado por la cinemática inversa, una vez que se conoce si esa configuración es alcanzable por el efector final pasará a ser una configuración valida para agarre de objetos.

Se propone el punto de agarre sobre el eje z del objeto.

$$P_g = (0, 0, L/2) \quad (4.43)$$

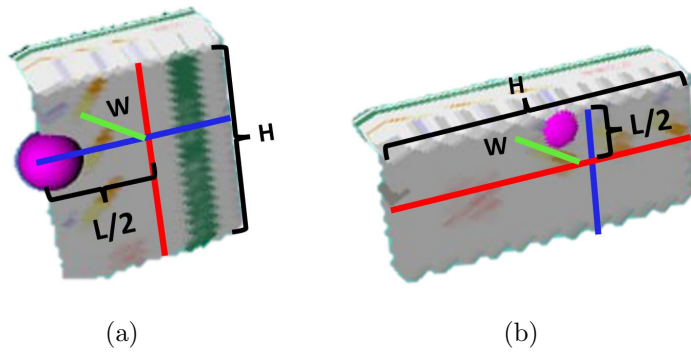


Figura 4.14: (a) El punto de agarre se genera en el borde lateral. (b) El punto de agarre se genera en el borde superior.

Generación de los candidatos de agarre

Se proponen candidatos de agarre sobre el punto de agarre propuesto. Estos candidatos se construyen realizando rotaciones sobre uno de los ejes coordenados del objeto un tamaño de paso $\Delta\theta$.

Sea la orientación del objeto expresada en los ángulos Raw , $Pitch$ y Yaw

$$Raw_{obj}, Pitch_{obj}, Yaw_{obj} \quad (4.44)$$

El candidato i -ésimo se genera realizando una rotación en $Pitch$ un tamaño de paso $\Delta\theta$.

$$C_i = \Delta\theta * i \quad (4.45)$$

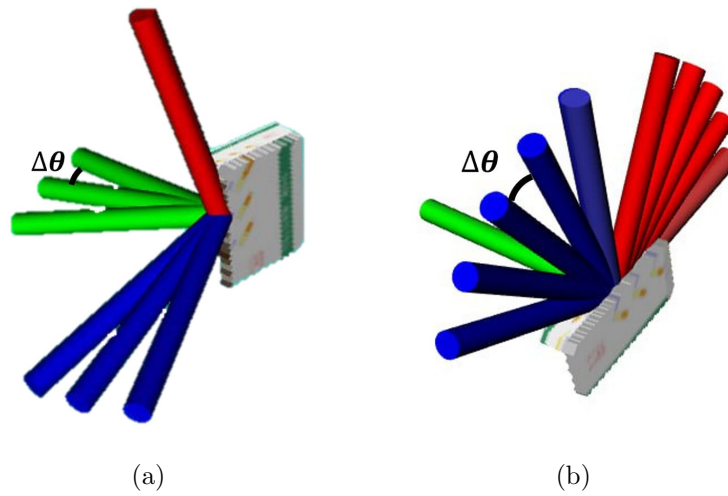


Figura 4.15: Rotaciones sobre el eje Y del objeto para la generación de candidatos de agarre con $\Delta\theta = -0,24rad$

4.8.9. Objeto esférico/cuboide

Agarres superiores son los más adecuados para este tipo de objetos pues su escasa altura produce centroides cercanos a la superficie y en consecuencia se tienen agarres peligrosos donde el efector final tiene altas probabilidades de chocar con el borde de la mesa y dañar su mecanismo.

Para conseguir agarres rápidos de objetos se proponen dos orientaciones a partir de las cuales se generan los candidatos. En la primera la pinza se orienta hacia enfrente 4.16b. En la segunda, la pinza se orienta de forma lateral 4.16a. Los candidatos se generan haciendo n rotaciones en *pitch* sobre los sistemas propuestos, cada tamaño de paso θ .

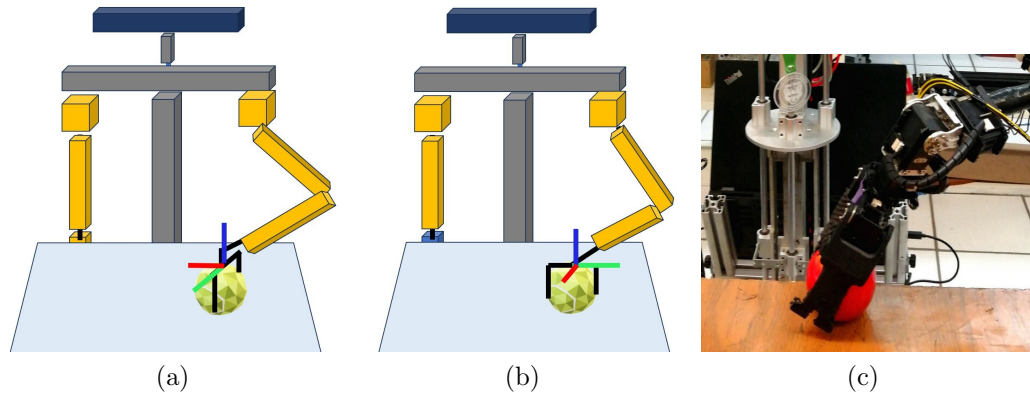


Figura 4.16: (a) Agarre con pinza orientada lateralmente. (b) Agarre con pinza orientada hacia enfrente. (c) Agarre de una manzana con la pinza orientada lateralmente.

4.9. Modelo Dinámico del Manipulador

Para representar el comportamiento físico del sistema cuando se encuentra sometido a fuerzas externas se utilizan modelos matemáticos que consisten en ecuaciones diferenciales no lineales. El sistema de n-GDL se puede considerar un sistema no lineal multivariable y relaciona variables de entrada y de salida.



Figura 4.17: La entrada consiste en pares y la salida son las posiciones y velocidades angulares de cada articulación del brazo.

A partir de las ecuaciones de Lagrange se obtienen las ecuaciones de la dinámica del manipulador. La ecuación 4.46 representa de forma compacta la dinámica de un manipulador de n-GDL, es una ecuación diferencial no lineal vectorial con los estados $[q^T \dot{q}^T]$.

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + g(q) = \tau \quad (4.46)$$

Donde

- $M(q)$ es la matriz de inercias de dimensión $n \times n$.
- $C(q, \dot{q})\dot{q}$ es el vector de fuerzas centrífugas y de Coriolis, de dimensión n .
- $g(q)$ es el vector de dimensión n de pares gravitacionales.
- τ es un vector de dimensión n llamado vector de pares externos y corresponde al los pares aplicados por los actuadores en las articulaciones.

El vector de pares gravitacionales $g(q)$ representa la fuerza de gravedad a la que se encuentran sometidos los manipuladores robóticos y depende solo de las posiciones articulares.

4.9.1. Control proporcional derivativo con compensación de gravedad(PD+)

Para un manipulador robotico, se puede lograr el objetivo de control de posición utilizando un control proporcional derivativo(PD), siempre y cuando el termino de gravedad $g(q) = 0$ se encuentre ausente del modelo del robot.

Cuando el vector de pares gravitacionales se encuentra dentro del modelo dinámico del sistema, se presentará un error de posición en estado estable que no podrá ser solventado por el control proporcional derivativo(PD). Un control PD con compensación de gravedad(PD+) es capaz de corregir el error de posición en estado estable producido por la fuerza de gravedad sobre cada eslabón del brazo, y para satisfacer el objetivo del control de posición. La ley de control del PD+ utiliza el vector de pares gravitacionales $g(q)$

$$\tau = K_p \tilde{q} + K_v \dot{\tilde{q}} + g(q) \quad (4.47)$$

K_p y $K_v \in \mathbb{R}^{n \times n}$ son matrices simétricas positivas definidas.

A diferencia del control PD, que no requiere conocer el modelo dinámico del robot, un control PD+4.47 necesita conocer de forma parcial el modelo del sistema, el conocimiento de $g(q)$.

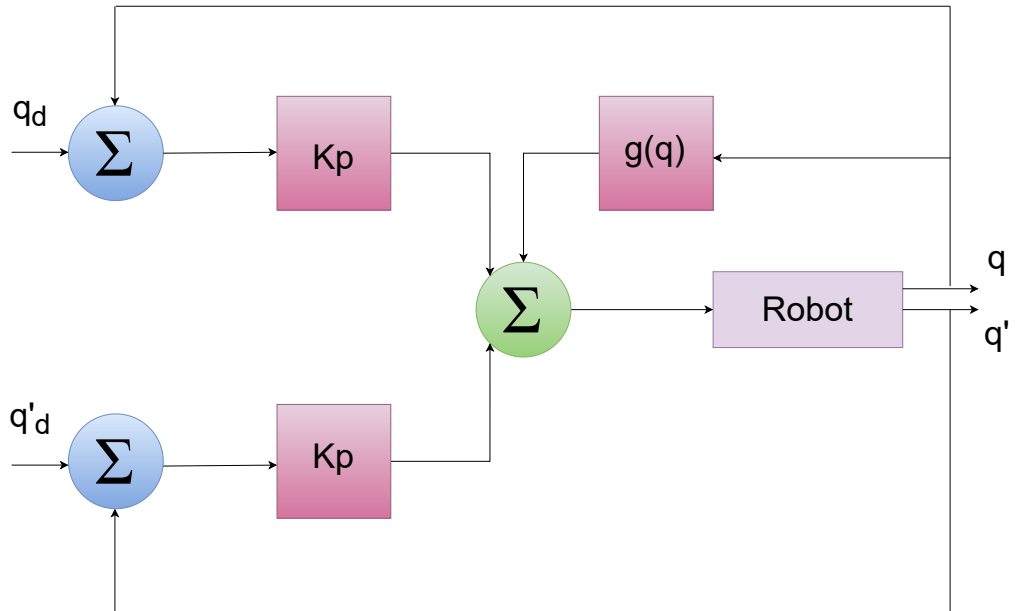


Figura 4.18: Diagrama de control PD+.

4.9.2. Implementación del controlador

La simulación del modelo se construye a partir del archivo de Formato Universal de Descripción de Robots (URDF por sus siglas en inglés ¹). Utilizando el paquete Robotics Systems de MATLAB. Esta toolbox brinda herramientas y algoritmos para diseñar, simular, probar y desplegar aplicaciones de robots móviles y manipuladores. Simscape; que permite crear rápidamente modelos de sistemas físicos dentro del entorno de Simulink, y ROStoolbox, que ofrece una interfaz que conecta MATLAB y Simulink con Robot Operating System (ROS y ROS 2). Con esta toolbox, se puede diseñar una red de nodos ROS y combinar los nodos ROS generados en MATLAB o Simulink con la red ROS existente. Para implementar la simulación del manipulador robótico se siguieron los pasos descritos en [27].

Las posturas inicial y final del manipulador se encuentran conectadas por una trayectoria basada en polinomios de 5 orden y generada con el bloque de trayectorias de Simulink. La ley de control 4.47 requiere que se calcule el

¹<http://wiki.ros.org/urdf>

vector de pares gravitacionales, para lo cual se utilizó el bloque de gravedad proporcionado por Robotics System Toolbox.

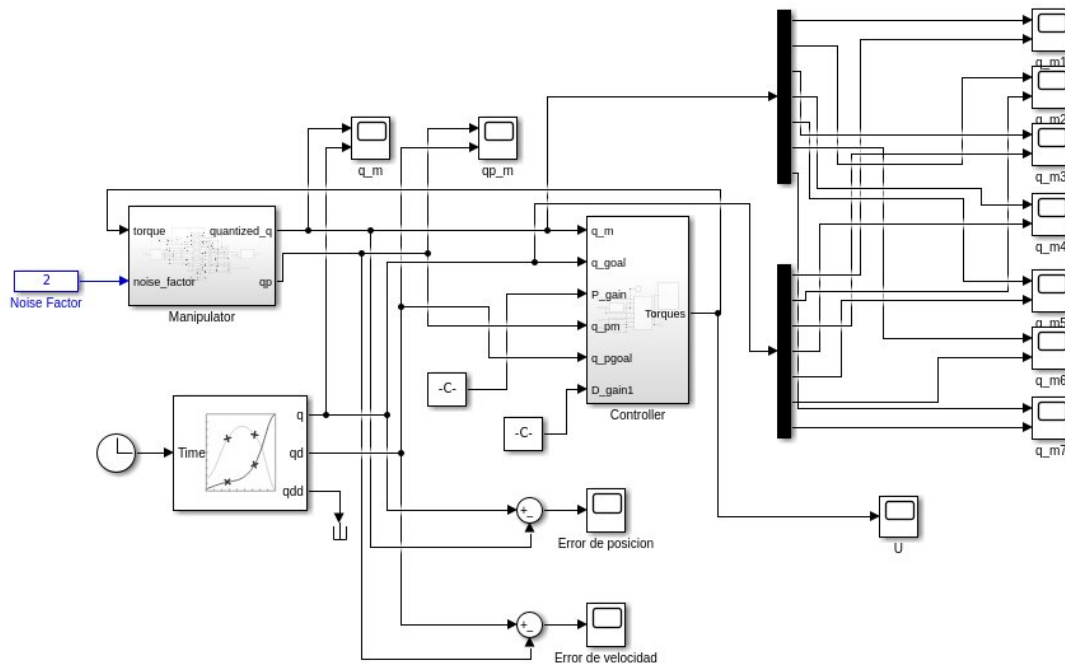


Figura 4.19: Diagrama del sistema.

En este capítulo se describió la cinemática directa e inversa, así como la cinemática diferencial del manipulador utilizado. Se describió como mejorar la convergencia de la solución a la ecuación de cinemática inversa. Como parte fundamental de la planeación de movimientos, se propusieron las reglas utilizadas para el agarre de objetos, basadas en la información obtenida de la nube de puntos, y el método seguido para la generación de candidatos. También se mostró la propuesta de un controlador proporcional derivativo con compensación de gravedad para disminuir el error de posición debido a la fuerza de gravedad que actúa sobre el manipulador. En el siguiente capítulo se aborda la implementación y los resultados obtenidos.

Capítulo 5

Resultados

En este capítulo se muestran los resultados de las pruebas realizadas al trabajo desarrollado en los capítulos anteriores con el manipulador robótico. Se realizaron pruebas de control de posición en simulación para el control Proporcional Derivativo con compensación de gravedad propuesto, y pruebas de agarre de objetos con el manipulador robótico con un conjunto formado por objetos cotidianos de diversas formas.

5.1. Implementación en ROS

El planeador de movimientos fue desarrollado en el sistema operativo de robot (ROS) y programado en python. ROS es un conjunto de bibliotecas de software y herramientas que ayudan a crear aplicaciones de robots. Además de que su código es abierto.¹

5.1.1. El simulador Gazebo

Gazebo es un simulador 3D multi-robot. Ofrece la posibilidad de simular con precisión y eficiencia, diversidad de robots, objetos y sensores en ambientes complejos interiores y exteriores. Gazebo genera, tanto la realimentación realista de sensores, como las interacciones entre los objetos físicamente plausibles, incluida una simulación precisa de la física de cuerpo rígido. Es un robusto motor de física, contiene gráficos de alta calidad, y las interfaces gráficas programáticas y convenientes. Gazebo es gratuito y de código abierto².

¹<https://www.ros.org/>

²<https://hemeroteca.unad.edu.co/index.php/publicaciones-e-investigacion/article/view/1593/1940>

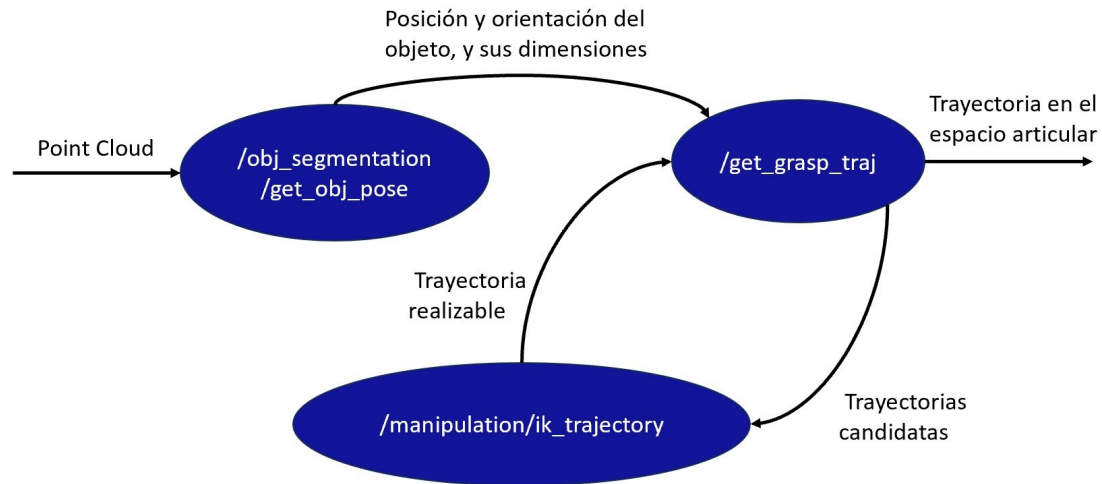


Figura 5.1: La nube de puntos del objeto se pasa como mensaje al nodo de segmentación donde se obtiene la posición y orientación del objeto, se le asocia un nuevo sistema coordenado y se obtienen las dimensiones aproximadas de su caja delimitadora, que es la entrada para el segundo nodo (mensaje de tipo "visionobject"). El resultado de este segundo nodo es un mensaje con una trayectoria que es una lista de configuraciones que conectan la actual con la más adecuada para el agarre del objeto.

5.1.2. Nodos de ROS creados

De acuerdo a la estructura de ROS, los nodos se comunican entre sí publicando mensajes en servicios. Un mensaje es una estructura de datos similar a las estructuras definidas en la programación en c. Los nodos fueron implementados en Python, adoptando la arquitectura de Nodo ROS procesal³.

La plataforma ROS permite que los nodos implementados funcionen de manera simultanea y compartan información entre ellos. El nodo de segmentación y pose contiene las funciones necesarias para realizar la segmentación por color implementada con el algoritmo k-medias, la detección de contornos de los objetos en la nube de puntos, el análisis de componentes principales y el sistema coordenado del objeto.

El nodo de agarre de objetos tiene como propósito planificar y generar la trayectoria que lleva al sistema coordenado del efector final desde su posición

³https://nu-mmr.github.io/navigation_site/lectures/node_structure.html

actual hasta a la posición de objeto, así como también se encarga de generar una configuración de agarre para la cual exista la cinemática inversa.

El nodo de segmentación se comunica con el nodo de agarre de objetos a través de mensajes personalizados que proporcionan información detallada del objeto como sus dimensiones aproximadas, forma geométrica primitiva aproximada y orientación respecto al sistema coordenado de la base del robot. Esta información es utilizada por el nodo de agarre de objetos para realizar la planeación de movimientos necesaria para manipular objetos.

El nodo de cinemática inversa fue modificado para obtener la solución de cinemática inversa para puntos distantes al efector final dentro del espacio de trabajo manteniendo el método de Newton Raphson.

5.2. Pruebas del control de posición

Para evaluar el controlador se eligió una postura típica del brazo en la manipulación de objetos(5.5) y donde la mayoría de las articulaciones tienen un cambio en su posición. El tiempo establecido para ir del punto inicial al punto final de la trayectoria fue de 3 segundos con un tiempo de 0.002 segundos entre cada punto de la trayectoria.

Se observó el comportamiento del control a través de gráficas que comparan posición deseada contra posición medida, así como también velocidad deseada contra velocidad obtenida, el error de posición, error de velocidad y la señal de control τ en el tiempo.

El análisis de los datos mostrados indica que, para el conjunto de gráficas ?? Los valores de cada articulación tendieron a la posición deseada para una trayectoria de 5 segundos. En conjunto con las gráficas de la imagen 5.7, se observó que para la posición deseada contra posición medida, el efecto de la compensación de gravedad realizado por el control PD+, el error en estado estacionario atribuido a este par, disminuye notablemente alcanzando errores de posición en cada articulación mucho menores a $0.01 \left[\frac{rad}{m} \right]$.

Una vez que el manipulador ha llegado a la posición deseada, la señal de control indica que para mantener esa configuración espacial, será necesario enviar un par constate de un valor determinado a cada articulación.

5.3. Pruebas de agarre de objetos

Antes de comenzar este proceso, se lleva al brazo a una posición de preparación en la que no hay puntos de singularidad y no es visible desde la cámara. Para demostrar la aplicabilidad real del sistema propuesto, se lle-

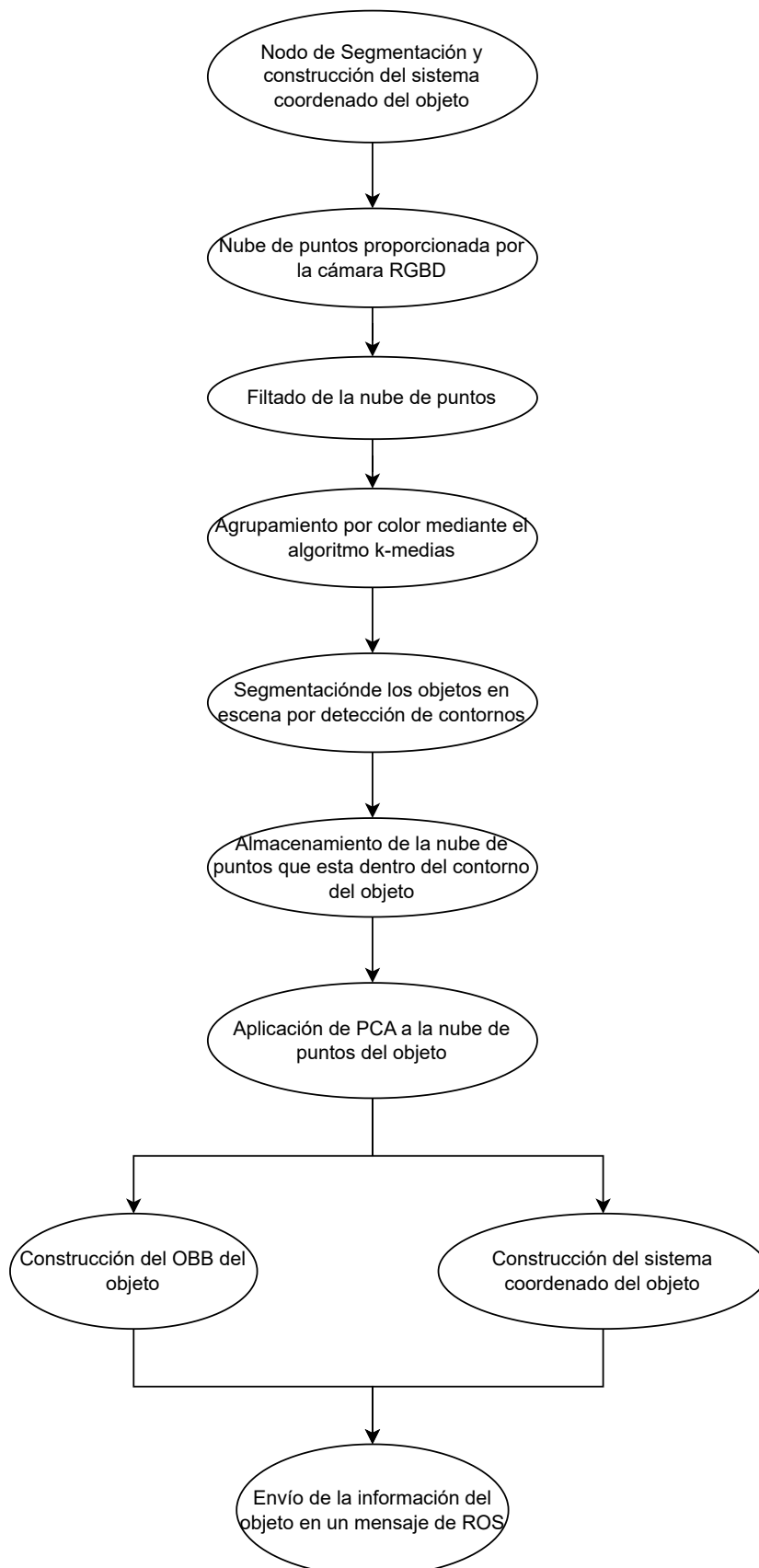


Figura 5.2: Diagrama del nodo de segmentación.

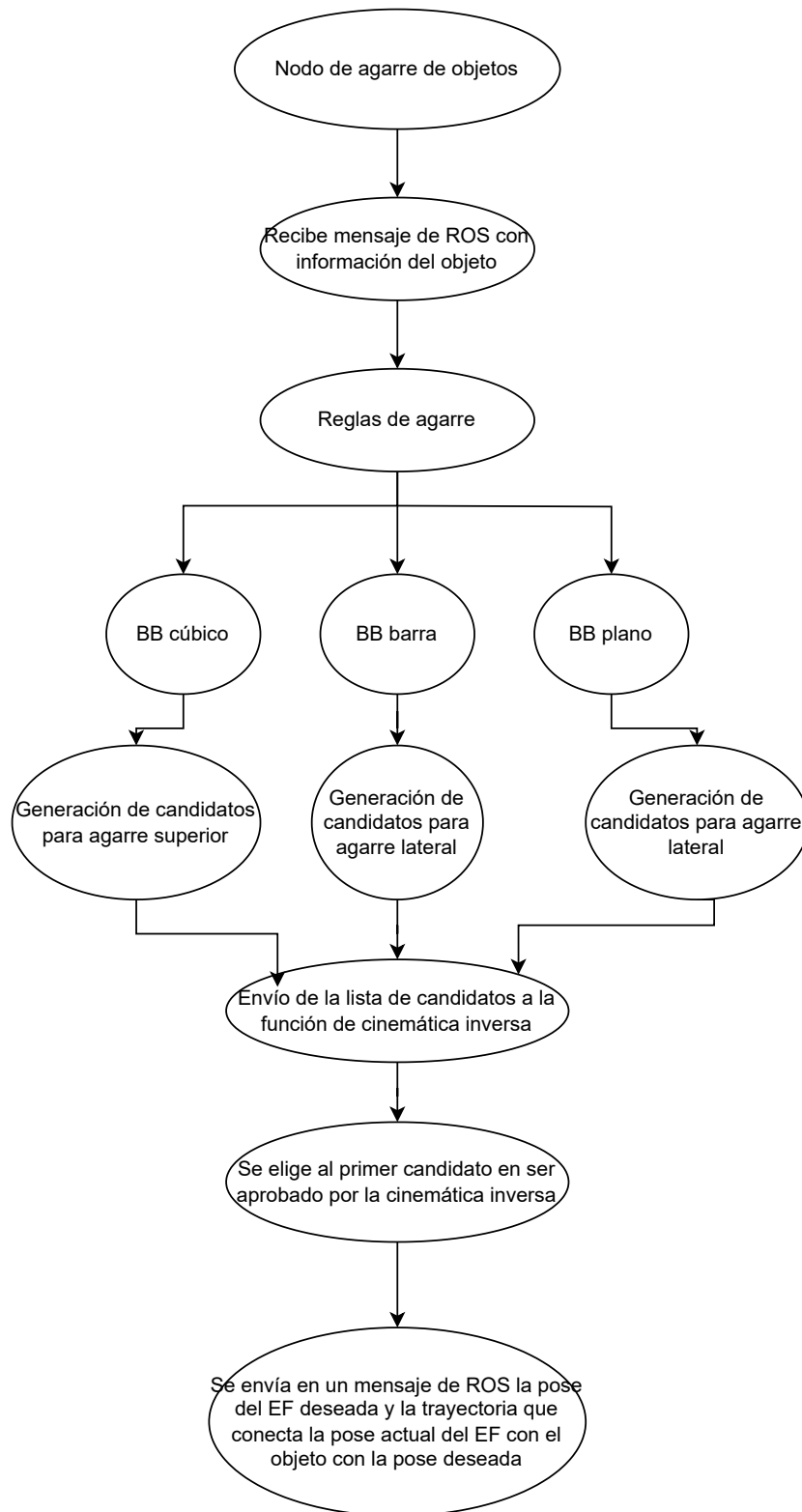


Figura 5.3: Diagrama del nodo de agarre.

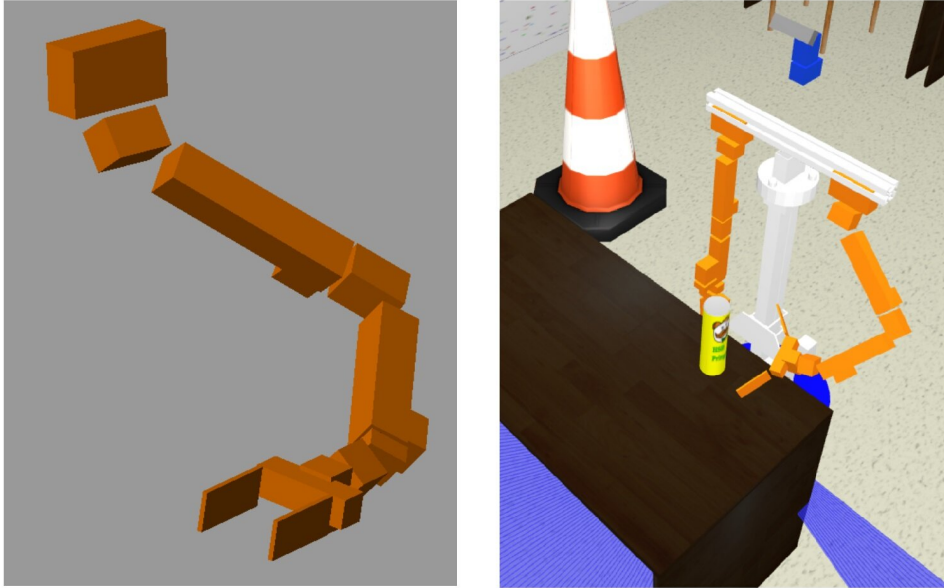


Figura 5.4: Postura típica del brazo robótico.

varon a cabo pruebas experimentales utilizando un total de 14 objetos(5.10) para desempeñar pruebas de agarre.

Se evalúa el sistema en conjunto con un reconocedor de objetos, estos objetos que han sido solicitados por el usuario se colocan en diferentes poses dentro de un grupo de objetos colocados al azar.

En cada experimento, el objetivo era generar configuraciones de agarre en la orientación del objeto. Se colocó sobre la mesa el objeto a reconocer, en una orientación y ubicación diferente cada vez y dentro del espacio de trabajo del manipulador.

Estas configuraciones incluían orientaciones en las que el objeto no era perfectamente paralelo ni perpendicular a la mesa, por lo que su eje principal tenía algún ángulo con respecto a la mesa. Para cada objeto y configuración probados, fueron realizados cinco intentos de agarre.

5.4. Integración del sistema a las tareas de servicio

Una vez que se probó el desempeño del planeador de acciones con distintos tipos de objetos cotidianos, fue integrado a las tareas de servicio del robot

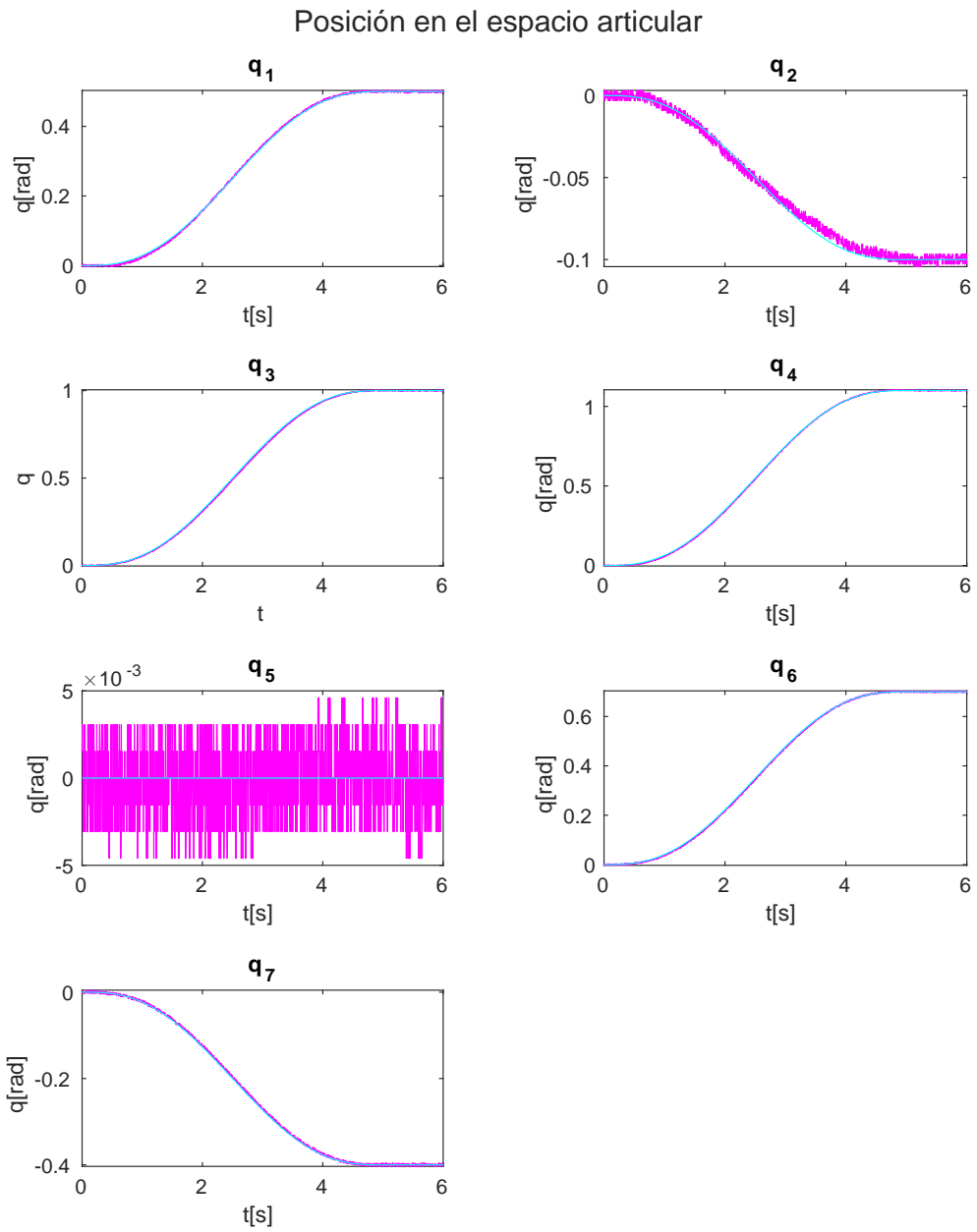


Figura 5.5: Gráficas para $q_1, q_2, q_3, q_4, q_5, q_6$ y q_7 , en azul la posición ideal y en magenta la posición medida.

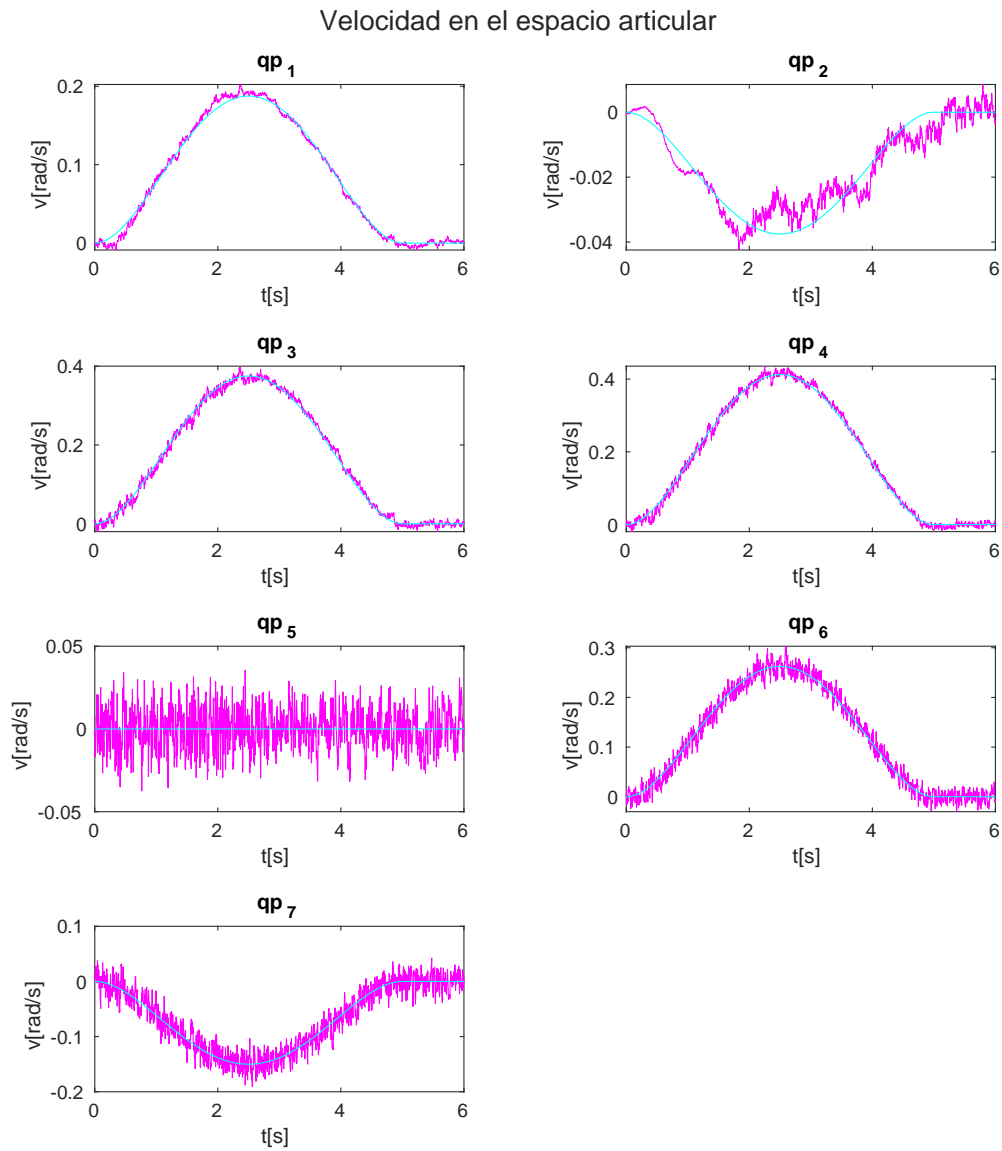


Figura 5.6: Gráficas para $q_{p1}, q_{p2}, q_{p3}, q_{p4}, q_{p5}, q_{p6}$ y q_{p7} , en azul la velocidad ideal y en magenta la velocidad medida.

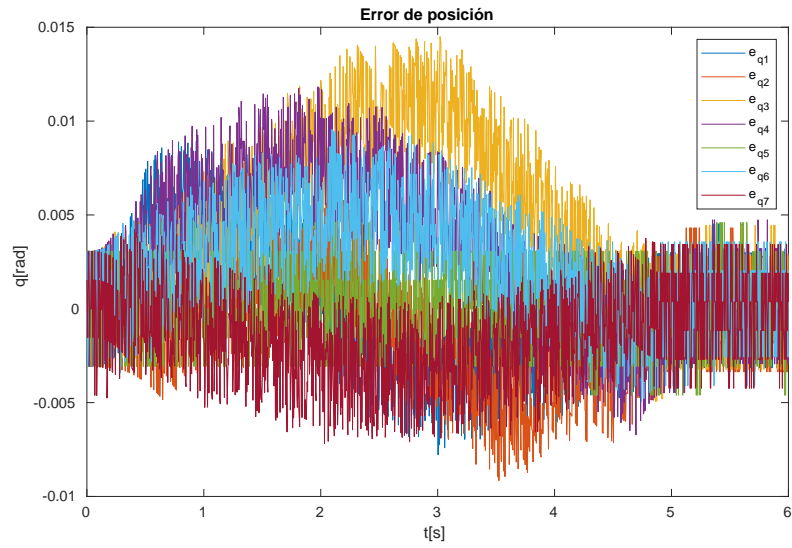


Figura 5.7: Error de posición para las articulaciones.

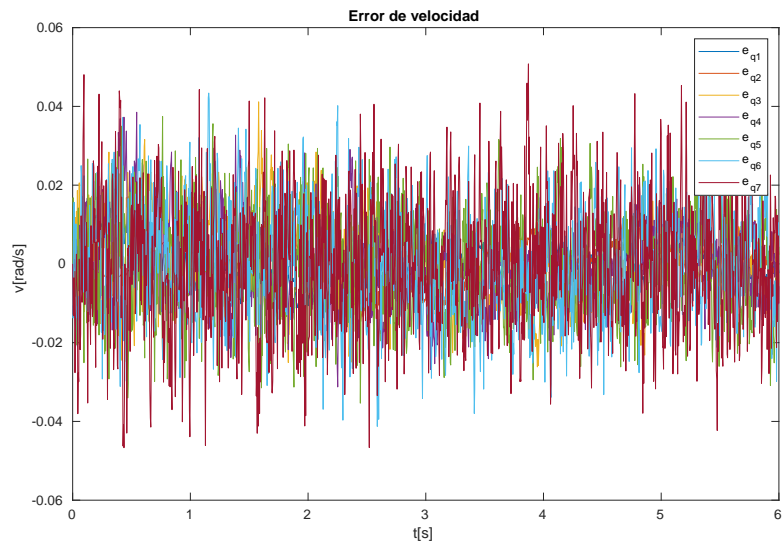


Figura 5.8: Error de velocidad para las articulaciones.

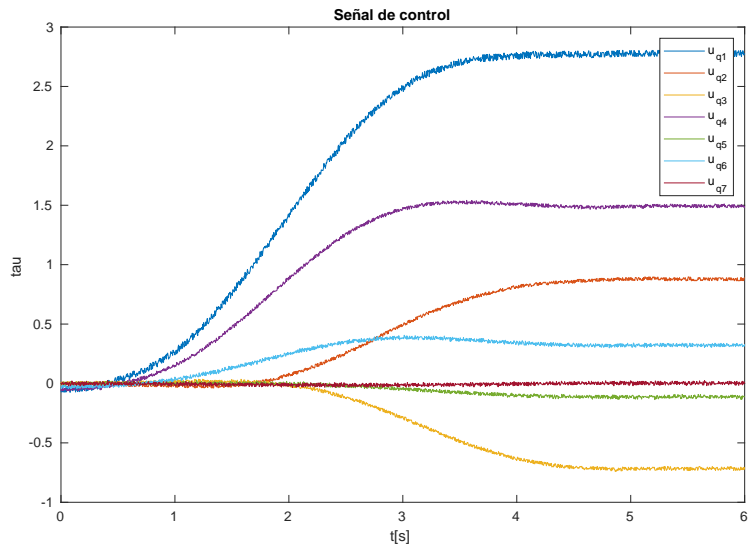


Figura 5.9: Señal de control τ .



Figura 5.10: Conjunto de objetos para las pruebas.

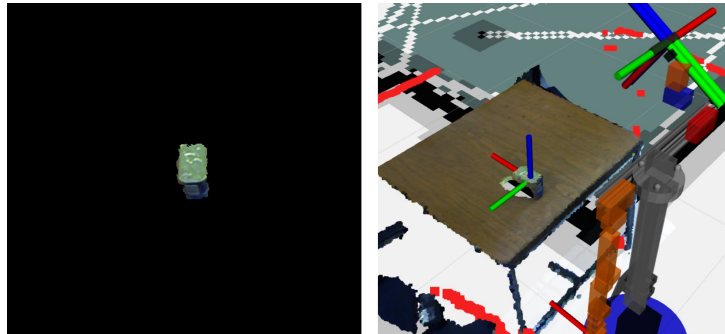


Figura 5.11: El eje x del sistema coordenado asociado al efector final es perpendicular a las pinzas de agarre.

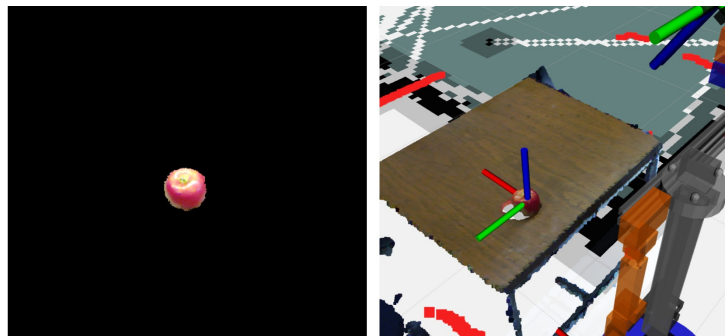


Figura 5.12: El eje x del sistema coordenado asociado al efector final es perpendicular a las pinzas de agarre.

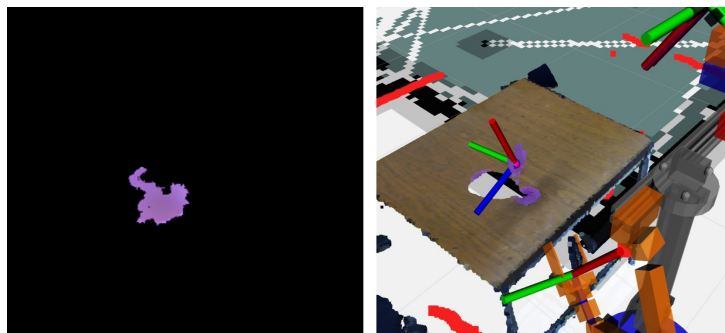


Figura 5.13: El eje x del sistema coordenado asociado al efector final es perpendicular a las pinzas de agarre.

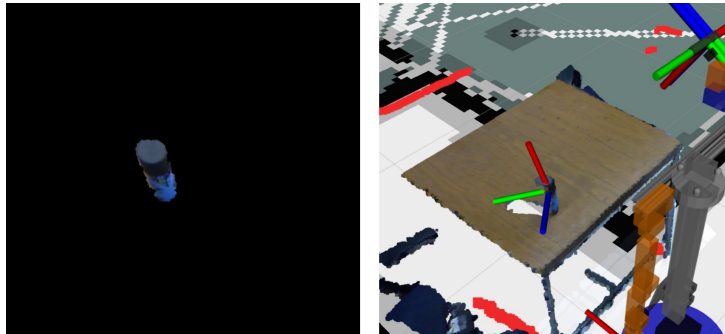


Figura 5.14: El eje x del sistema coordenado asociado al efector final es perpendicular a las pinzas de agarre.

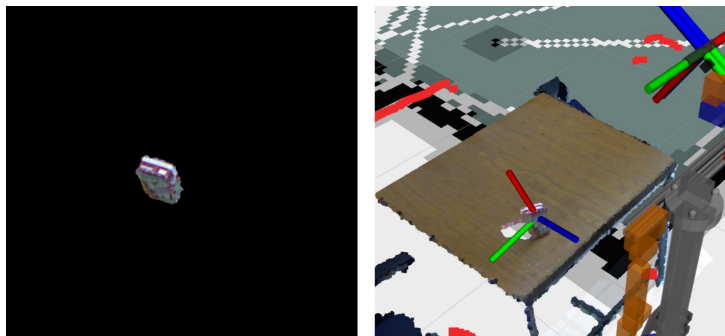


Figura 5.15: El eje x del sistema coordenado asociado al efector final es perpendicular a las pinzas de agarre.

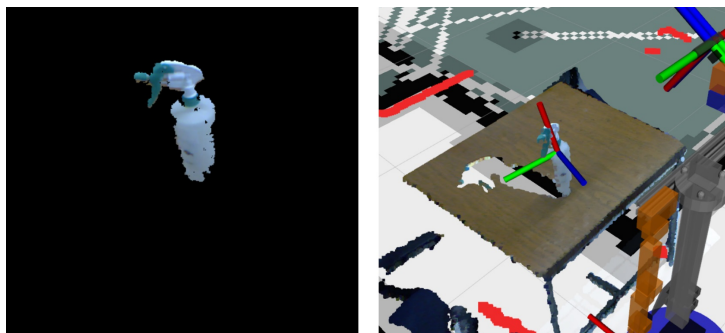


Figura 5.16: El eje x del sistema coordenado asociado al efector final es perpendicular a las pinzas de agarre.

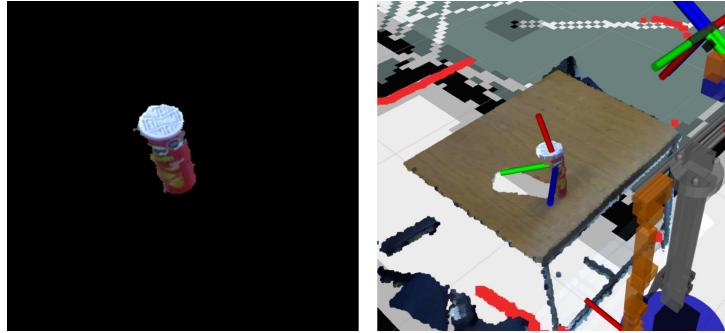


Figura 5.17: El eje x del sistema coordenado asociado al efector final es perpendicular a las pinzas de agarre.

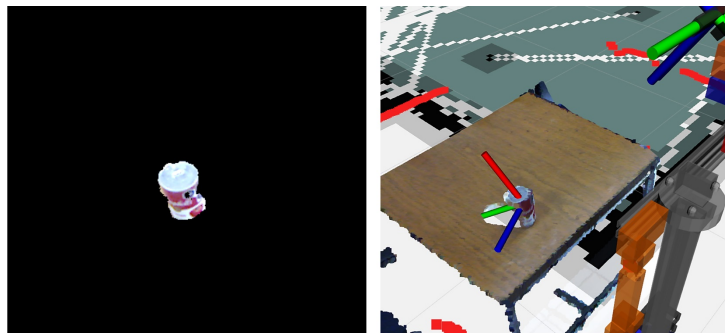


Figura 5.18: El eje x del sistema coordenado asociado al efector final es perpendicular a las pinzas de agarre.

Tipo	Objeto	Dimensiones [m]	Intentos	tipo de agarre	Agarres exitosos	tasa %
prismático	pringles	0.2x0.08x0.08	5	lateral	3	60
prismático	pringles	0.2x0.08x0.08	5	superior	5	100
caja	jugo	0.1x0.05x0.03	5	superior	4	80
cubico	taza	0.06x0.07x0.06	5	superior	3	60
prismático	lata de refresco	0.07x0.07x0.06	5	superior	4	80
esferoide	manzana	0.07x0.06x0.06	5	superior	3	60
caja	caja de cereal	0.3x0.2x0.05	5	lateral	4	80
prismático	lata de conservas	0.012x0.07x0.07	5	superior	4	80
box	jamón en lata	0.07x0.12x0.06	5	superior	4	80
prismático	copa	0.23x0.09x0.09	5	lateral	4	80
cuboide	pera	0.08x0.06x0.05	5	superior	4	80
prismático	mostaza	0.13x0.07x0.04	4	lateral	3	60
prismático	desinfectante	0.25x0.1x0.1	5	lateral	5	100
prismático	desodorante	0.1x0.05x0.05	5	superior	3	60

Tabla 5.1: Tasa de agarres para distintos objetos del dataset.

mediante una maquina de estados(5.19), en la cual, en con junto con otras habilidades se logra la realizacion de tareas con las cuales brinda servicio a los humanos.

5.4.1. Pruebas para la liga Robocup@Home

La liga RoboCup@Home tiene como objetivo desarrollar tecnología de robots de servicio y asistencia. Es la competencia anual internacional más grande para robots de servicios autónomos. Se utiliza una serie de pruebas de referencia para evaluar las capacidades y el rendimiento de los robots en un entorno doméstico realista y no estandarizado⁴.

Entre las pruebas que involucran la manipulación de objetos y donde puede ser utilizado el sistema desarrollado, consisten en el transporte de equipaje (Carry My Luggage), manipulación de objetos para elaborar un platillo(Serve Breakfast) o el transporte de alimentos para clientes dentro de

⁴<http://www.robocup.org/domains/3>

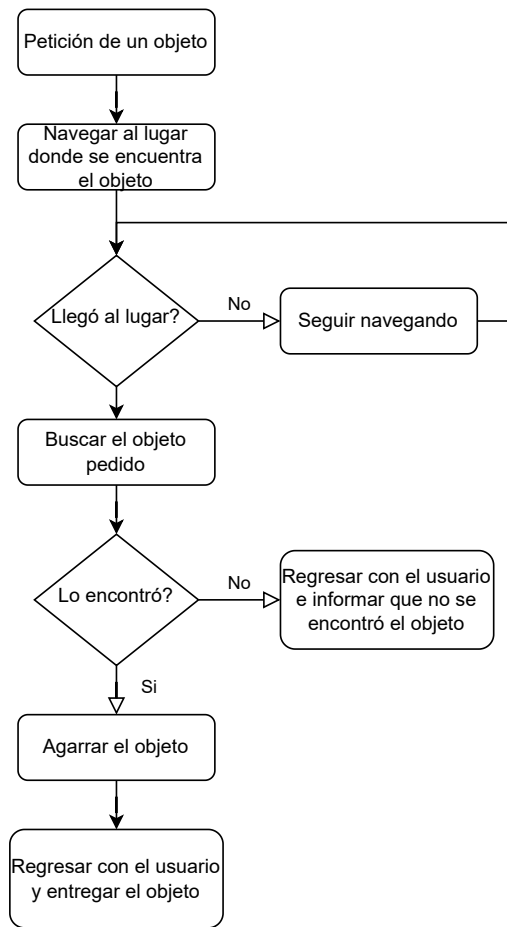


Figura 5.19: Ejemplo de una maquina de estados para atender a un usuario que solicita un objeto ubicado en una locación distante.

un restaurante(Restaurant).

En este capítulo se evaluó el desempeño del sistema para agarrar objetos implementado y del control proporcional derivativo con compensación de gravedad(PD+). Se describió la forma en que la que fue distribuido el trabajo en la plataforma ROS. Se mostraron los resultados de las pruebas experimentales realizadas en simulación para el control PD+, a través de gráficas de posición y velocidad, y los resultados de las pruebas de agarrar realizadas en la plataforma real se mostraron a través de una tabla de datos e imágenes donde se mostraron las etapas del planeador de acciones.

El siguiente capítulo aborda las conclusiones derivadas del análisis de los resultados obtenidos y el trabajo a futuro que podría realizarse.

Capítulo 6

Discusión

6.1. Conclusiones

Se desarrolló e implementó un sistema de planeación de movimientos para el agarre de objetos en un manipulador robótico. Se validó mediante pruebas experimentales y se obtuvieron poses de agarre funcionales en la mayoría de las pruebas.

El análisis de componentes principales brinda una buena aproximación de las dimensiones de la caja delimitadora del objeto y contribuye al agarre rápido de objetos desconocidos sin un alto coste computacional. Los candidatos de agarre generados en tiempo real incrementaron las probabilidades de un agarre funcional del objeto.

Combinar el uso de métodos numéricos con el uso de trayectorias polinomiales para la resolución de la función de cinemática inversa aumenta notablemente la cantidad de configuraciones que el manipulador puede tomar dentro de su espacio de trabajo y repercute en el desempeño del sistema para el agarre de objetos.

El control proporcional derivativo con compensación de gravedad ayuda a reducir notablemente el error de posición del manipulador robótico debido al par de gravedad.

Una limitación del método propuesto fue que la generación de trayectorias se diseñó para un espacio de trabajo libre de obstáculos, reportándose en las pruebas de agarre, colisiones del efector final con el borde de la mesa antes de completar tomar el objeto, por lo cual tuvo que modificarse la posición de pre-agarre a una donde el sistema coordinado del efector final se encontrara a mayor altura que la mesa para solucionar el problema.

6.2. Trabajo futuro

Para mejorar el sistema desarrollado, el trabajo futuro podría abordar el desarrollo de algoritmos para generar trayectorias para la evasión de obstáculos con el fin de que el robot pueda tomar el objeto sin que el efector final colisione con el borde de la mesa o con demás objetos que se encuentren alrededor del objeto que se desea tomar. También podrían ser desarrolladas medidas de la calidad de agarre en el algoritmo del planeador, para disminuir las probabilidades de un deslizamiento del objeto. En cuanto a la velocidad del sistema, podría ser entrenado con una red neuronal para mejorar los tiempos de agarre.

Bibliografía

- [1] R. I. B. *Apuntes de métodos numéricos*. Facultad de Ingeniería, Ciudad Universitaria, México, D.F, primera edición. edition, 1983.
- [2] T. Braunl. *EMBEDDED ROBOTICS*. SPRINGER, 2008.
- [3] O. K. Bruno Siciliano. *Handbook of Robotics*. Springer, 2016.
- [4] R. L. Burden and J. D. Faires. *Análisis numérico /*. Cengage Learning ;, México :, septima edición. edition, 2011. Traducción de: Numerical analysis. 9th ed. 2011.
- [5] S. C. Chapra. *Métodos numéricos para ingenieros /*. McGraw Hill,, Mexico :, séptima edición. edition, 2015.
- [6] L. Cruz, D. Lucio, and L. Velho. Kinect and rgb-d images: Challenges and applications. In *2012 25th SIBGRAPI conference on graphics, patterns and images tutorials*, pages 36–49. IEEE, 2012.
- [7] B. Cyganek. *AN INTRODUCTION TO 3D COMPUTER VISION TECHNIQUES AND ALGORITHMS*. John Wiley and Sons, John Wiley and Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom, 2009.
- [8] J. P. David A. Forsyth. *COMPUTER VISION, A modern approach*. Prentice Hall., 2012.
- [9] L. S. d. G. Eduardo Solar González. *Apuntes de Algebra Lineal*. Departamento de matemáticas básicas, Facultad de Ingeniería, UNAM, 1985.
- [10] J. F. Engelberger. *ROBOTICS IN SERVICE*. SPRINGER SCIENCE+BUSINESS MEDIA, B.V., 120 Pentonville Road, London N1 9JN, 1989.

- [11] D. R. Faria, P. Trindade, J. Lobo, and J. Dias. Knowledge-based reasoning from human grasp demonstrations for robot grasp synthesis. *Robotics and Autonomous Systems*, 62(6):794–817, 2014.
- [12] A. K. Gary Bradski. *Learning OpenCV*. O’Reilly Media, Inc., Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472, 2008.
- [13] H. Hanafusa, T. Yoshikawa, and Y. Nakamura. Analysis and control of articulated robot arms with redundancy. *IFAC Proceedings Volumes*, 14(2):1927–1932, 1981. 8th IFAC World Congress on Control Science and Technology for the Progress of Society, Kyoto, Japan, 24-28 August 1981.
- [14] S. H. Howie Choset, Kevin Lynch. *Principles of Robot Motion*. The MIT Press, 2005.
- [15] S.-Q. Ji, M.-B. Huang, and H.-P. Huang. Robot intelligent grasp of unknown objects based on multi-sensor information. *Sensors (Basel, Switzerland)*, 19, 2019.
- [16] I. Jolliffe and I. Jolliffe. Generalizations and adaptations of principal component analysis. *Principal Component Analysis*, pages 223–234, 1986.
- [17] S. Kathiv. *Handbook of Robotics*. Springer, 2008.
- [18] A. Koubaa. *Robot Operating System(ROS), The complete Reference*. Springer, 2016.
- [19] A. O. Kurt Demaagd. *Practical Computer Vision with SimpleCV*. O’Reilly Media, Inc., 2012.
- [20] S. M. LaValle. *Planning algorithms*. Cambridge University Press, 2006.
- [21] Q. Lei, G. Chen, and M. Wisse. Fast grasping of unknown objects using principal component analysis. *Aip Advances*, 7(9):095126, 2017.
- [22] B. S. Lorenzo Sciavicco. *Modelling and Control of Robot Manipulators*. Springer, 2000.
- [23] L. V. Lorenzo Sciavicco. *Robotics*. Springer, 2009.
- [24] K. M. Lynch and F. C. Park. *MODERN ROBOTICS MECHANICS, PLANNING, AND CONTROL*. Cambridge University Press, 2017.

- [25] M. V. M. Spong, S. Hutchinson. *Robot Modeling and Control*. JOHN WILEY & SONS, INC, 2006.
- [26] D. Marr. *VISION*. W.H Freeman and Company., 1982.
- [27] M. Negrete, J. Savage, and J. Avendaño. Mass estimation in manipulation tasks of domestic service robots using fault reconstruction techniques. *arXiv preprint arXiv:2010.06116*, 2020.
- [28] P. Rivera, E. Valarezo Añazco, and T.-S. Kim. Object manipulation with an anthropomorphic robotic hand via deep reinforcement learning with a synergy space of natural hand poses. *Sensors*, 21(16):5301, 2021.
- [29] D. Rodriguez and S. Behnke. Transferring category-based functional grasping skills by latent space non-rigid registration. *IEEE Robotics and Automation Letters*, 3(3):2662–2669, 2018.
- [30] I. Sainul, S. Deb, and A. K. Deb. A novel object slicing based grasp planner for 3d object grasping using underactuated robot gripper. In *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, volume 1, pages 585–590. IEEE, 2019.
- [31] D. F. Sebastian Thrun, Wolfram Burgard. *Probabilistic Robotics*. Stanford University, 2000.
- [32] L. Shapiro. *Computer Vision*. The University of Whashington, 2000.
- [33] B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer, 2000.
- [34] B. S. Zapata-Impata, P. Gil, J. Pomares, and F. Torres. Fast geometry-based computation of grasping points on three-dimensional point clouds. *International Journal of Advanced Robotic Systems*, 16(1):1729881419831846, 2019.