



UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ

CENTRO DE INVESTIGACIÓN Y ESTUDIOS DE POSGRADO

FACULTAD DE INGENIERÍA

Teleoperación unilateral asimétrica de un robot industrial por medio de un dispositivo móvil

TESIS PROFESIONAL

PARA OBTENER EL GRADO DE:

MAESTRO EN INGENIERÍA ELÉCTRICA

OPCIÓN EN CONTROL AUTOMÁTICO

Presenta:

Ing. Gabriela Rangel Ramírez

Asesores de tesis:

Dr. Emilio González Galván

Dr. Ambrocio Loredo Flores

San Luis Potosí, S.L.P., Febrero de 2012

UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERÍA
CENTRO DE INVESTIGACIÓN Y ESTUDIOS DE POSGRADO
MAESTRÍA EN INGENIERÍA ELÉCTRICA

Tesis:

"Teleoperación unilateral asimétrica de un robot industrial por medio de un dispositivo móvil"

Presenta:

Ing. Gabriela Rangel Ramírez

Sinodales:

Dr. Emilio J. González Galván
Asesor

Dr. Ambrocio Loredo Flores
Co-Asesor

Dra. Elvia Palacios Hernández
Revisor

Dr. Andrei Yurievich Gorbatchev
Revisor

Dr. Daniel Ulises Campos Delgado
Revisor

Agradecimientos

A mis padres, hermanos y familia por darme todo su apoyo.

A los doctores Emilio González y Ambrocio Loredo por compartirme sus conocimientos y darme la oportunidad de ser parte de su equipo de investigación.

A mis amigos de toda la vida, gracias por su amistad.

A mis amigos y compañeros de generación que estuvieron conmigo a lo largo de esta etapa, y la hicieron más llevadera.

A mis compañeros de laboratorio por el intercambio de ideas y las pláticas amenas a lo largo de este proyecto.

A los maestros del CIEP por todas las asesorías brindadas.

Al CONACyT por la beca otorgada .

Resumen

El presente trabajo de tesis se enfoca en la implementación de un sistema asimétrico teleoperado unilateralmente para el control de un robot industrial.

El desarrollo principal de este trabajo consiste en el diseño de una aplicación para un dispositivo móvil que permita, de manera ergonómica e intuitiva, designar la trayectoria que debe seguir un manipulador industrial. El proyecto considera la optimización de la pose del robot, a través de la maximización del índice de manipulabilidad.

La presentación de resultados obtenidos ha sido estructurada en tres etapas: la primera etapa consiste en la validación del sistema teleoperado, describiendo el retardo de comunicación y el correcto seguimiento del esclavo respecto al maestro.

La segunda etapa consiste en la optimización de la posición del robot industrial durante una trayectoria, donde se describen los problemas relacionados con las posiciones no alcanzadas por el manipulador y la forma como fueron solucionadas.

Finalmente, la última etapa, consiste en la integración de un control de impedancia de posición dentro de un esquema de teleoperación unilateral, desarrollado de tal manera que permite un adecuado desempeño en tareas de interacción teleoperadas.

Índice general

Índice de figuras VIII

Índice de tablas XI

Introducción 1

1. Estado del Arte de la Teleoperación 4

1.1. Componentes de un sistema teleoperado 5

1.2. Telemanipuladores en el sector nuclear 6

1.3. Telemanipuladores para vehículos submarinos 8

1.4. Telemanipuladores en el sector espacial 10

1.5. Telemanipuladores en el sector médico 13

2. Cinemática del robot industrial 16

2.1. Modelo Cinemático Directo FANUC M16iB-20T 17

2.2. Modelo Cinemático Inverso FANUC M16iB-20T 20

2.3. Jacobiano del Manipulador 23

2.3.1. Velocidad Angular 24

2.3.2. Velocidad Lineal 25

2.4. Manipulabilidad 25

2.4.1. Índice de Manipulabilidad 26

2.4.2. Mecanismo de 2 grados de libertad 27

3. Características del dispositivo móvil **29**

3.1. Desarrollo de aplicaciones para el Ipod Touch 30

3.2. Pantalla Táctil 32

3.3. Acelerómetros 34

3.4. Interfaz 34

4. El problema de asimetría cinemática **39**

4.1. Control Cinemático 39

4.2. Obtención de la última variable de control 40

4.3. Propuesta de la Función Objetivo 42

4.4. Optimización de la Función Objetivo 44

4.4.1. Búsqueda de la sección dorada 45

5. Resultados experimentales **50**

5.1. Plataforma experimental 50

5.2. Trayectoria Preplaneada 52

5.3. Optimización de Posición 55

5.4. Implementación de control cinemático de impedancia basado en posición 60

5.4.1. Variación de parámetros de impedancia 64

Conclusiones y trabajo futuro **68**

A. Control de impedancia **70**

A.1. Control de Impedancia Basado en Posición 70

B. Fotografías **72**

C. Códigos de Programa **75**

Bibliografía **89**

Índice de figuras

1.1. Principales avances en la teleoperación, agrupados en sectores de aplicación 4

1.2. Esquema para el enfoque de la teleoperación 6

1.3. Presentación del robot Handyman, manipulando un aro de hula hula 7

1.4. Vehículo submarino Victor 6000 10

1.5. Prototipo del sistema manipulador remoto de la estación espacial (SSRMS) 13

1.6. Robot Zeus para cirugías cardíacas 14

1.7. Robot DaVinci para cirugías cardíacas 15

2.1. Esquema para la cinemática directa 17

2.2. Esquema para el cálculo de la cinemática inversa 22

2.3. Mecanismo planar de 2 grados de libertad 27

3.1. Dispositivos móviles de Apple 29

3.2. Estrategia de diseño MVC 31

3.3. Partes que componen la pantalla del Ipod Touch 32

3.4. Algoritmo de reconocimiento 33

3.5. Dirección de los ejes coordenados en el Ipod Touch 34

3.6. Valores ideales de aceleración para diferentes orientaciones del dispositivo 35

3.7. Evolución de interfaces para el dispositivo maestro 36

3.8. Interfaz diseñada para el sistema teleoperado 37

3.9. Dirección de la inclinación del dispositivo móvil para el control de la orientación 38

4.1. Sistema Coordenado fijo en la base del robot 40

4.2. Orientación del robot, debida a la última variable de control 41

4.3. Rotación de un ángulo α alrededor del eje coordenado x 42

4.4. Variación del índice de manipulabilidad por la variable α 44

4.5. Condiciones necesarias para la sección dorada 46

4.6. Comienzo del algoritmo de búsqueda de la sección dorada 47

4.7. Algoritmo para la búsqueda de la sección dorada 49

5.1. Sistema teleoperado implementado 52

5.2. Trayectoria del manipulador FANUC M16-iB 20T 53

5.3. Tiempo de muestreo de una trayectoria 53

5.4. Trayectoria perpendicular al eje z 54

5.5. Tiempos de muestreo de una trayectoria con diferente taza de desplazamiento 55

5.6. Índice de manipulabilidad de una trayectoria planeada 56

5.7. Comparación del índice de manipulabilidad para una trayectoria 57

5.8. Trayectoria del manipulador sobre el eje z 57

5.9. Índice de manipulabilidad optimizado 58

5.10. Valor del ángulo α durante una trayectoria 58

5.11. Cambios del índice de manipulabilidad en una trayectoria definida 59

5.12. Corrección del valor del ángulo α optimizado 61

5.13. Corrección del índice de manipulabilidad optimizado 61

5.14. Algoritmo en el controlador del robot 62

5.15. Escenario para el control de impedancia 63

5.16. Trayectoria preplaneada del sistema teleoperado 64

5.17. Valores de las fuerzas de contacto durante la interacción con la rampa 65

5.18. Variación del parámetro de rigidez $\mathbf{K}_d$ 65

5.19. Variación del parámetro de amortiguamiento $\mathbf{B}_d$ 66

5.20. Variación del parámetro de inercia $\mathbf{M}_d$ 66

B.1. Robot Fanuc Fanuc LR Mate 200iC 72

B.2. Robot Fanuc M16 iB 20T 73

B.3. Unidad de control del Fanuc M16 iB 20T 73

B.4. MacBook utilizada para la programación del dispositivo móvil 74

B.5. Dispositivo móvil Ipod Touch utilizado para el sistema teleoperado 74

Índice de tablas

2.1. Características del Fanuc M16iB-20T 16

2.2. Dimensiones del robot Fanuc M16iB-20T 18

2.3. Dimensiones de la herramienta 20

4.1. Posición inicial del manipulador 43

5.1. Valores estadísticos de la comunicación del sistema teleoperado 53

5.2. Valores estadísticos de la posición del eje z, sobre una trayectoria simple 54

5.3. Valores estadísticos de la comunicación del sistema teleoperado 55

5.4. Tabla comparativa de valores optimizados 60

Introducción

Desde siempre, el hombre ha desarrollado distintas herramientas para poder aumentar su capacidad de manipulación. Estos desarrollos llegaron finalmente a lo que se conoce como sistemas de teleoperación maestro-esclavo, en los que un manipulador denominado esclavo reproduce fielmente los movimientos de un dispositivo maestro, controlado a su vez, manualmente, por un operador humano; se puede decir que es entonces cuando la teleoperación cobra importancia como tecnología. Este trabajo se centrará en los sistemas robóticos teleoperados.

En el desarrollo de robots teleoperados se involucran varias ramas de la robótica como la electrónica, los sistemas de comunicación, el control, la programación avanzada, la mecánica, entre otros [4], [5], [6], [7]. Para el caso del presente trabajo, se busca relacionar el tema de la teleoperación con el campo de la computación ubicua. Mark Weiser, en Septiembre de 1991, describió su visión sobre este campo, en el que uno de los principales objetivos es disimular la presencia de los dispositivos computacionales colocándoles en un segundo plano. Este objetivo de crear dispositivos que se mezclen en la vida cotidiana hasta que lleguen a ser indistinguibles, supone una potencial revolución que puede hacer cambiar el modo de vida diaria. Las personas se centrarán en las tareas que deben hacer, no en las herramientas que utilizan, porque se pretende que esas herramientas pasen desapercibidas.

El significado de enviar la computación a un segundo plano, está referido a dos conceptos diferentes pero relacionados. El primero es el significado literal de que la tecnología de la computación se debe integrar en los objetos, tareas y entornos cotidianos. La segunda es que esta integración se debe realizar de forma tal que la introducción de computación en estos objetos no interfiera con las actividades para las que son usadas, y que siempre proporcionen un uso más cómodo, intuitivo y útil.

Estos objetos cotidianos en los que se integra la tecnología de la computación pasan a tener una serie de propiedades que permiten la creación del entorno ubicuo buscado. Algunas de estas propiedades son: Comunicación entre dispositivos, memoria, sensibilidad al contexto y que interactuen con el usuario. Otro factor que ha incrementado el desarrollo de proyectos de teleoperación, es la tecnología con la que cuentan

los sistemas de comunicación actual [8]-[13].

Hoy en día existen muchos proyectos de investigación sobre computación ubicua, tanto en las universidades como en las empresas: Oxygen en el MIT, Aura en el CMU, CoolTown en HP. Cada uno de estos proyectos se centra en diferentes aspectos de la computación ubicua, y persiguen diferentes objetivos tanto a largo como a corto plazo, pero todos ellos intentan hacer de la computación ubicua una realidad [1],[2],[3].

Motivación

La mayoría de los robots industriales aún son programados usando el mismo proceso de aprendizaje, utilizando la terminal de enseñanza o teach pendant. Programar y controlar un robot industrial a través de la terminal de enseñanza del robot es una tarea incómoda y complicada, puesto que requiere de una gran experiencia técnica. Por lo tanto se requieren nuevas y más intuitivas formas para programar uno de estos dispositivos.

Sin embargo, el teach pendant continúa siendo la entrada común del robot que da acceso a todas las funciones que brinda el sistema, como mover el manipulador, crear y editar programas, etc. En los últimos años los diseñadores de robots han hecho grandes esfuerzos por hacer amigables al usuario los teach pendants, implementando interfaces intuitivas como programas basadas en iconos, pantallas touch a color, palancas en tercera dimensión, un mouse de 6 dimensiones, o también desarrollando teach pendant inalámbricos. A pesar de esto, permanece la dificultad, el aburrimiento y cansancio de operar el robot desde este dispositivo, especialmente para usuarios no expertos .

En este proyecto se propone la sustitución del teach pendant por medio de un dispositivo móvil, específicamente uno fabricado por la empresa Apple, conocido comercialmente como Ipod Touch. El proyecto plantea exclusivamente la tele operación unilateral en el que las señales de posición y orientación, obtenidas del dispositivo móvil, son enviadas al robot esclavo, un robot industrial FANUC M-16iB 20T, en el que se establecerán condiciones cinemáticas específicas para el desarrollo de la maniobra, buscando optimizar una condición cinemática particular, por ejemplo, el índice de manipulabilidad.

Se debe tener en cuenta, para el desarrollo de la interfaz, las características específicas de un ambiente industrial, los requerimientos de seguridad, confiabilidad y ligereza. Además, que sea entendible por diferentes usuarios, no solo por especialistas, de modo que una persona inexperta sea capaz de interactuar naturalmente con ella y que se satisfagan ciertos criterios de seguridad, como los que se tienen disponibles en las terminales de enseñanza.

Existen ciertas configuraciones cinemáticas del robot, denominadas singularidades, que originan un desempeño pobre del dispositivo que se traducen en: la operación excesiva de los motores, la imposibilidad de realizar localmente algunos movimientos o una baja precisión en el posicionamiento del efector final. Entonces, al optimizar la posición del robot, se minimiza la probabilidad de caer en alguna singularidad

evitando un desempeño pobre del mecanismo.

Contribuciones del trabajo

Los objetivos fueron cubiertos en su totalidad y entre las contribuciones generadas directamente de este proyecto de tesis se pueden mencionar las siguientes:

- Diseño de una interfaz intuitiva y ergonómica para el control de la posición y orientación de un robot industrial, utilizando las herramientas que brinda el dispositivo móvil.
- Propuesta de un control cinemático para optimizar la posición de un robot respecto al eje unido a la punta de la herramienta al seguir una trayectoria deseada.
- Implementación de un sistema de comunicación inalámbrico para comunicar al robot con el dispositivo móvil.
- Diseño de un algoritmo de control cinemático en lenguaje nativo del robot (KAREL) para la optimización de la posición del mismo, capaz de ser incluido en cualquier controlador FANUC .
- Integración con un algoritmo de control de interacción.

Organización del trabajo

Los principales elementos que componen un sistema teleoperado se presentan en el Capítulo 1. En este capítulo también se muestra el desarrollo de las plataformas de teleoperación desde sus inicios en los años cuarenta, hasta la actualidad, en diversos sectores de servicios.

En el Capítulo 2 se obtiene el modelado cinemático del manipulador FANUC M-16iB 20T, que consta de la cinemática directa, la cinemática inversa y el Jacobiano. Todos estos conceptos son indispensables para la simulación y control de la posición del robot como se plantea en el objetivo de este proyecto. En el Capítulo 3 se describen las herramientas con las que dispone el dispositivo móvil, así como la forma especial de programación y el diseño de la interfaz para lograr una interacción intuitiva con el operador.

En la mayoría de los sistemas teleoperados existe una asimetría entre los dispositivos; en el Capítulo 4 se aborda este problema, de manera que se propone la optimización de una función objetivo por medio de un método numérico para corregir este problema.

En el Capítulo 5 se muestran los resultados experimentales de la implementación del sistema teleoperado entre un Ipod Touch y un robot industrial. En este proyecto se tuvo la oportunidad de contar con dos robots industriales para la etapa experimental, los manipuladores M-16iB 20T y LR Mate 200iC de la marca FANUC.

Capítulo 1

Estado del Arte de la Teleoperación

Para obtener una visión global de las plataformas de teleoperación actuales, es conveniente disponer, en primer lugar, de unas referencias históricas acerca del desarrollo de la teleoperación. La Figura 1.1 muestra los principales logros de la teleoperación en las últimas décadas. Estos avances se encuentran clasificados según los sectores en los que han sido aplicados, lo que da una perspectiva global de la evolución de la teleoperación, así como de las aplicaciones que la han impulsado. En los últimos años se han entrelazado los avances tecnológicos producidos en los diferentes sectores, a la vez que han surgido nuevos campos de aplicación [27].

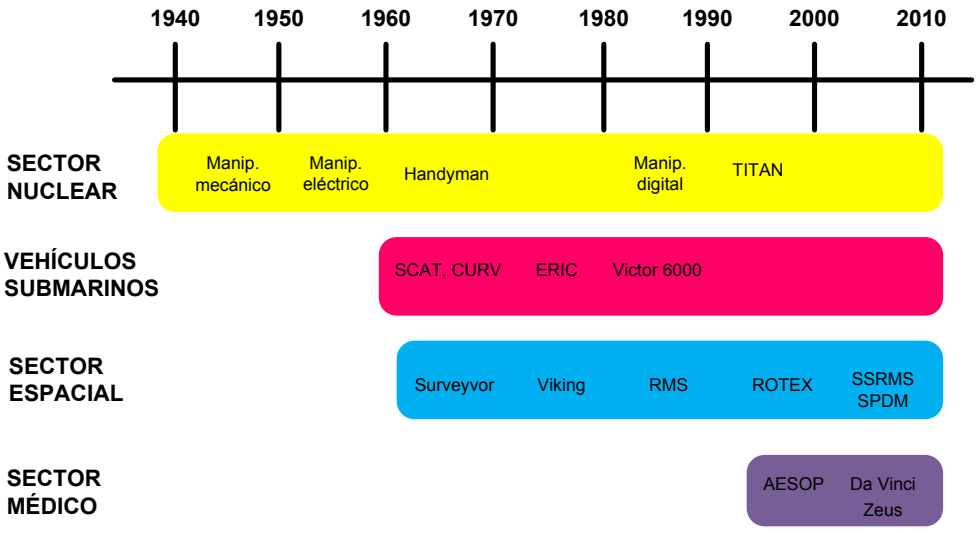


Figura 1.1. Principales avances en la teleoperación, agrupados en sectores de aplicación

1.1. Componentes de un sistema teleoperado

En cualquier esquema de teleoperación se distinguen necesariamente los siguientes elementos:

- Zona local, en la cual el operador interactua con un dispositivo maestro.
- Zona remota, en la cual el dispositivo esclavo interactua con el entorno remoto.
- Un operador el cual es un ser humano que realiza, a distancia, el control de la operación. Su acción puede ir desde un control continuo hasta una intervención intermitente.
- Interfaz, es un conjunto de dispositivos que permiten la interacción del operador con el sistema de teleoperación; éste se considera como parte del manipulador maestro.
- Dispositivo teleoperado o esclavo. En este caso, cualquier dispositivo robótico puede ser utilizado como esclavo en un esquema de teleoperación, si bien la gran parte de los mismos disponen de una arquitectura de control cerrada que impide el acceso a bajo nivel de los actuadores del robot. Esta característica conduce a que la mayoría de las plataformas de teleoperación se encuentran implementadas a nivel de comandos del robot.
- Control y canales de comunicación. Conjunto de dispositivos que modulan, transmiten y adaptan el conjunto de señales de información que se transmiten entre la zona remota y la local. En la literatura se encuentran ejemplos que van desde plataformas que utilizan protocolos estándar de comunicaciones (como TCP/IP), entre tareas de tiempo real que se ejecutan en la zona remota y en la zona local, hasta soluciones en las cuales una única tarea de tiempo real controla maestro y esclavo, que se encuentran comunicados eléctricamente.
- Sensores. Son un conjunto de dispositivos que recogen la información, tanto en la zona local como de la zona remota, para ser utilizada por la interfaz y el control, por ejemplo los encoders de los motores.

Además de los elementos anteriores, las dificultades esbozadas, relativas a los problemas de sincronización de las tareas de control en el entorno local y remoto, unidas a la creciente capacidad de cómputo y modelado de sistemas mecánicos, están motivando un gran interés por los simuladores virtuales, bien como sustitutos de la zona remota, bien como ayuda en las tareas de teleoperación.

Como se mencionó anteriormente, cualquier dispositivo robótico puede ser utilizado como esclavo en una arquitectura de teleoperación. Sin embargo, es importante resaltar las diferencias que existen en cuanto al control de los mismos, desde el mismo momento en el que la figura del operador aparece en el esquema de control. Eso ha hecho que los robots utilizados como esclavos hayan sido, en su mayoría, diseñados específicamente para dicho propósito. Sin embargo, recientemente aparecen artículos de experimentación con robots industriales a los cuales primeramente ha habido que acceder a su sistema de control. Además, los

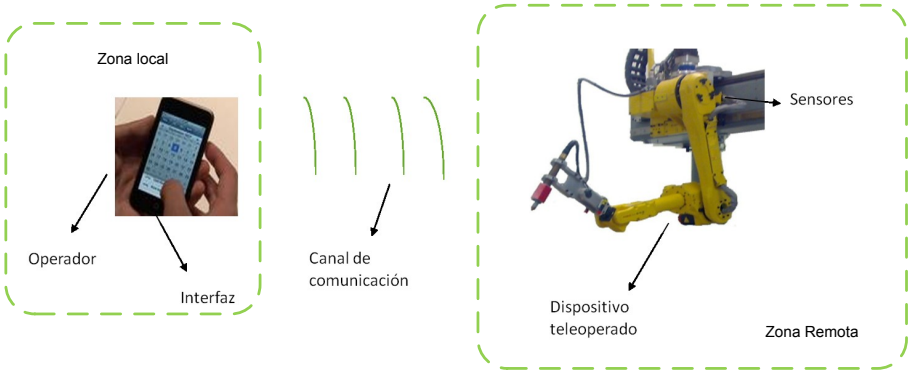


Figura 1.2. Esquema para el enfoque de la teleoperación

robots industriales están siendo utilizados en arquitecturas de teleoperación en las cuales la comunicación se realiza a nivel de comandos del robot.

Utilizando los diferentes sectores industriales y de servicio que aparecen en la Figura 1.1, se muestra a continuación un breve resumen de los telemanipuladores utilizados en diferentes campos de teleoperación [30].

1.2. Telemanipuladores en el sector nuclear

Durante la época de la guerra fría, las investigaciones desarrolladas entorno a la energía nuclear motivaron la aparición de los primeros teleoperadores, destinados a la manipulación de elementos radiactivos sin riesgo para el operador. Como se muestra en la primera fila de la Figura 1.1, a finales de los años 40 se desarrollaron los primeros mecanismos que permitían trabajar sobre elementos peligrosos a distancia, garantizándose de este modo la seguridad del operador.

En la actualidad, los robots se continúan utilizando para la manipulación de probetas y residuos radiactivos, habiéndose incorporado además a tareas de inspección y mantenimiento en instalaciones con contaminación radiactiva, como las centrales nucleares, así como tareas de intervención en caso de accidentes (Three Mille Island 1976, Chernobil 1986).

Los primeros desarrollos de interés fueron realizados en 1948. Raymond Goertz diseñó el primer brazo articulado teleoperado en el Laboratorio Nacional de Argonne (ANL, por sus siglas en inglés), para la Comisión de Energía Atómica. El diseño se basa enteramente en el acoplamiento mecánico entre el maestro y los brazos de esclavos (con cables de acero y poleas). Este primer prototipo puede ser considerado como un manipulador mecánico avanzado. Dicho dispositivo constaba de diferentes articulaciones que permitían transmitir los movimientos del operador, que se encontraba en un extremo, hacia el otro extremo en el que se hallaba una pinza, que manipulaba directamente los objetos.

La continuación de estos trabajos dio como resultado, en 1954, el primer telemanipulador maestro-esclavo de accionamiento eléctrico y con reflexión de fuerza. Este prototipo si se considera un verdadero telemanipulador. Constaba de un brazo articulado de accionamiento eléctrico, denominado esclavo, cuyo controlador recibía órdenes de movimiento para cada una de sus articulaciones. Las órdenes eran generadas por el operador a través de un maestro, sobre el que se reflejaba la fuerza que era realizada durante la manipulación de los objetos. Posteriormente se desarrollaron nuevos dispositivos más sofisticados, que disponían de realimentación visual mediante cámaras de televisión dando lugar a los modelos E3 en 1985 y E4 en 1963. A



Figura 1.3. Presentación del robot Handyman, manipulando un aro de hula hula

finales de los años cincuenta, la compañia General Electric estaba en la necesidad de un manipulador, especialmente delicado para manejar experimentos con un motor de avión atómico. Ralph Mosher, un ingeniero mecánico, desarrolló un sistema de teleoperación denominado Handyman. En los sistemas teleoperados anteriores, el manipulador esclavo era muy rígido por lo cual, para este sistema, se planteó la pregunta de ¿por qué un ser humano es tan eficiente y un robot esclavo tan torpe?. La respuesta a esta interrogante fue que al manipulador esclavo le faltaba una percepción de contacto. Esto se solucionó con una retroalimentación de fuerza.

Handyman fue construido entre 1958-1959 en Schnetectady, Nueva York, con una unidad de control de accionamiento hidroeléctrico. Contaba de dos brazos articulados en cuyos extremos se localizaban unas pinzas de cuatro grados de libertad cada una, lo suficientemente sensibles para empacar huevos, lo suficientemente fuertes como para aplastar pelotas de golf, y hábil como para encender un fósforo. El maestro utilizado, con un aspecto muy aparatoso, tenía forma de exoesqueleto rodeando los brazos del operador.

También, durante los sesenta, se desarrolló el manipulador Mascot, realizado por ENEA en Italia, con una configuración similar a la del E3. Avances sobre este prototipo dieron lugar a modelos de altas prestaciones, como el Mascot IV utilizado por el Jet Propulsion Laboratory para diferentes tareas de mantenimiento.

A partir de estos desarrollos aparecieron nuevos manipuladores con mejores características, que poco a poco incorporaron el uso de microprocesadores, aumentando significativamente sus capacidades. En Francia, Jean Vertut y Carl Flatau desarrollaron, durante los años 70, los telemanipuladores MA-22 y MA-23. Estos trabajos fueron realizados por la Agencia de Energía Atómica Francesa (CEA por sus siglas en francés).

En los 80, el ORNL (Oak Ridge National Laboratory), en colaboración con Central Research Laboratories, desarrolló el primer telemanipulador totalmente digital denominado M2. Éste constaba de una avanzada y compleja interfaz para el operador, en la que destacaba la retroalimentación visual mediante cámaras de visión y la reflexión de fuerza sobre los maestros. Este manipulador y los siguientes derivados de él, han sido diseñados para la manipulación de residuos radioactivos de las centrales nucleares, dentro del programa de gestión de residuos de combustible nuclear del Departamento de Energía de los Estados Unidos de Norteamérica.

En los años 90 han sido desarrolladas nuevas familias de telemanipuladores con prestaciones avanzadas. Ejemplo de ello es el telemanipulador Titanic de Schilling, desarrollado en los noventa, y muy utilizado en aplicaciones para la manipulación de materiales radiactivos. Otro ejemplo, en Europa, es la empresa Telerob que desde principios de los noventa fabrica telemanipuladores y vehículos para el manejo de residuos nucleares.

1.3. Telemanipuladores para vehículos submarinos

El entorno submarino es otro de los ambientes hostiles y arriesgados donde un robot puede aportar importantes ventajas, evitando riesgos a los seres humanos debido a las extremas condiciones de presión, temperatura, corrientes de agua, etc., en las que operan.

Dos aplicaciones han sido las que en el pasado han motivado el mayor número de desarrollos de robots submarinos. Los tendidos de cable para comunicaciones y las plataformas oceánicas para la extracción de petróleo y gas. No obstante, en la actualidad hay otras aplicaciones que se benefician del uso de robots en el fondo marino. Dentro de ellas pueden citarse: la investigación oceánica ya sea en sus facetas biológicas ó geológicas, las aplicaciones de tipo militar, especialmente centradas en la vigilancia, localización y neutralización o en su caso una recuperación de minas o armamento hundido, la localización y rescate de barcos y aviones hundidos, y la inspección y reparación de buques o de construcciones con estructura sumergida (presas, puentes, etc.).

El origen de los primeros vehículos submarinos [29], incorporando telemanipuladores, fue debido a la

Armada de los EE. UU., los cuales eran guiados desde la superficie por un buque. Sus primeras misiones fueron la recolección de los torpedos. Posteriormente, pasaron a ser utilizados para el rescate de restos de accidentes, como los restos de naufragios de barcos o accidentes aéreos.

El programa desarrollado para la realización de estos vehículos comenzó en los años 60. Los primeros en aparecer fueron los denominados Snoopy y el Teleoperador Sumergible Accionado por Cable (SCAT por sus siglas en inglés, Sumersible Cable-Actuated Teleoperator). Portaban un solo manipulador con dos o tres grados de libertad, e incorporaban una o dos cámaras de vídeo. Tenían como misión el acopio de torpedos y la inspección del fondo marino.

Posteriormente, apareció el Vehículo de Recuperación Submarina Controlada por Cable (CURV por sus siglas en inglés, Cable Underwater Recovery Vehicle). Estos alcanzaban una mayor profundidad de trabajo, hasta 600 metros, y disponían de hasta dos manipuladores con más grados de libertad que los anteriores, lo que facilitaba la realización de los trabajos submarinos. En Julio de 1966, España se vio involucrada en el empleo de un robot submarino con fines militares. El robot CURV fue utilizado para localizar y recuperar la bomba H, hundida a pocos kilómetros de la costa de la localidad de Palomares en Almería, como consecuencia de la colisión de 2 aviones de las fuerzas aéreas de los EEUU. Tras varios intentos frustrados de realizar la localización y recuperación con submarinos tripulados, finalmente fue el CURV quien realizó el rescate, utilizando sus telemanipuladores.

En Francia también fueron desarrollados vehículos submarinos, para la observación y realización de trabajos submarinos por la armada francesa. Estos trabajos comenzaron en los años 70 con el ERIC 1 y el ERIC II; incorporando este último dos telemanipuladores MA-23 capaces de trabajar a una profundidad máxima de 6000 m. También se desarrollaron vehículos de uso específico, como los MNV (Mine Neutralization Vehicle) para la recogida de minas, y los NP (Nozzle Plug) para la recolección de los cohetes de la NASA.

Durante los años 80 se lanzó un programa más ambicioso por el que se pretendía el desarrollo de vehículos capaces de hacer trabajos de cierta complejidad en el fondo del mar, WSP (Work System Package-Conjunto de Sistemas de Trabajo). A éstos se les dotó de dos telemanipuladores, en algunos casos un tercer brazo auxiliar, sistemas avanzados de visión, y diferentes tipos de herramientas para la realización de los trabajos submarinos. El robot Victor 6000 es un robot submarino desarrollado por el IFREMERV (Institut Francais de I'Explotation de la Mer), especialmente diseñado para misiones de exploración de fondos oceánicos hasta una profundidad de 6000 metros. Se une al barco nodriza, desde el que se le teleopera mediante un cable de un alcance máximo de 8500 metros.

Para las misiones de localización e inspección de barcos, así como de estructuras sumergidas (puentes, plataformas, etc.), existen pequeños robots submarinos como el SEeaOtter que está equipado con 4 motores de propulsión, 2 motores para moverse hacia delante y atrás y 2 motores para moverse vertical y horizontal. Está equipado con una cámara de televisión que permite transmitir imágenes de vídeo a la superficie, donde pueden ser grabadas. El operario tiene la posibilidad de controlar, mediante un sistema JoyStick, el movimiento del robot submarino y la orientación de la cámara y sistema de iluminación. Un robot submarino

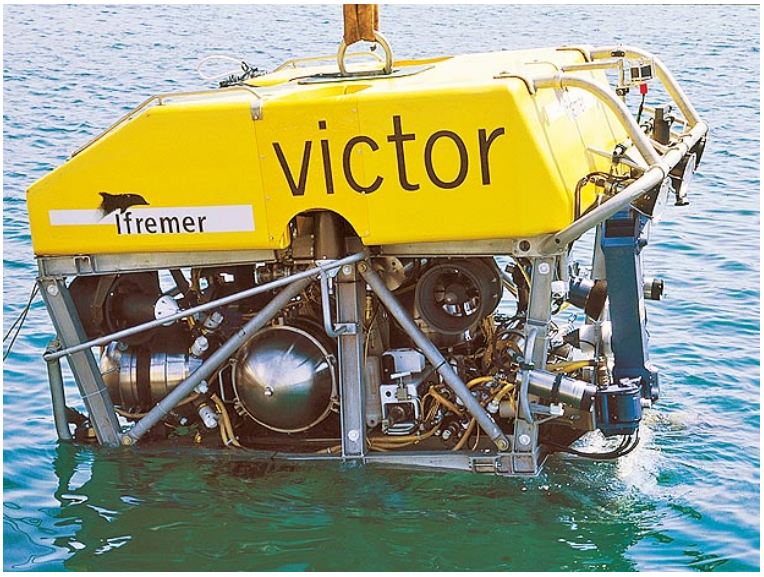


Figura 1.4. Vehículo submarino Victor 6000

similar a éste, pero con una capacidad de inmersión superior, el Jason Jr, fue utilizado conectado a un minisubmarino tripulado (el Alvin) a principio de 1986 para inspeccionar el Titanic hundido a 3800 metros de profundidad.

En la actualidad, se están desarrollando numerosos vehículos para diferentes tipos de aplicaciones. Como ejemplo de ello están los sistemas destinados al mantenimiento de tuberías submarinas. La empresa japonesa Tokio Electric Power Inc. junto a Mitsubishi Heavy Industries Ltd., desarrolló un robot para la limpieza de tuberias submarinas de centrales eléctricas. En el Reino Unido, un consorcio de empresas relacionadas con las perforaciones petrolíferas del Mar del Norte, están desarrollando un vehículo submarino con un telemanipulador guiado por un maestro para la realización de trabajos de inspección y limpieza de tuberías.

1.4. Telemanipuladores en el sector espacial

Los avances en el sector espacial han favorecido significativamente la evolución de la teleoperación, dando lugar al desarrollo de nuevos telerobots y numerosos trabajos de investigación. Los principales aspectos estudiados son la aplicación de robots teleoperados a tareas espaciales y la influencia que tienen los retardos en las comunicaciones en la estabilidad de los sistemas teleoperados.

El uso de telemanipuladores en el sector espacial ha tenido un gran éxito debido a la aportación de importantes ventajas en la realización de misiones para astronautas e incluso siendo éstos la única solución posible en muchos casos. De manera general, las ventajas que aporta el uso de los robots en el espacio pueden resumirse como siguen: incremento de la seguridad, debido al ambiente hostil que brinda el espacio;

incremento de la productividad, el tiempo se aprovecha de manera más efectiva; incremento de la fiabilidad y la posibilidad de realizar tareas imposibles para el ser humano.

La NASA clasifica las misiones de los robots en el espacio en tres grupos:

- Robots extravehiculares (EVR): Son robots que deben realizar misiones en el exterior de naves en órbita terrestre, como la puesta en órbita o rescate de satélites, ensamblado y mantenimiento de la estación espacial internacional (ISS), etc. El ejemplo más representativo de este tipo de robots es el Remote Manipulator System (RMS), construido por la empresa canadiense SPAR en 1981 y que ha efectuado numerosas misiones a bordo del transbordador espacial.
- Robots para exploración de otros planetas: se trata de robots móviles, con frecuencia dotados de capacidades de manipulación, que han sido y son utilizados para la exploración de la superficie de otros planetas. A este tipo pertenecen las primeras realizaciones, (Surveyvor, Lunakhods, Viking), así como el más reciente Sojourner.
- Robots para tareas intravehiculares: se trata de robots, de caracteristicas no muy diferentes a la de los robots industriales, incorporados a las naves con el objeto de realizar experimentos científicos en entornos controlados. Así, el desarrollo de experimentos físicos, químicos y biológicos, etc. en condiciones de microgravedad, puede ser preprogramado en tierra con la debida precisión y reproducidos en órbita sin precisar de la atención de los astronautas.

A excepción de este último grupo de tareas, los robots espaciales suelen ser teleoperados, bien por la tripulación de la nave que los transporta, o bien, desde la Tierra. En este segundo caso, las limitaciones en la calidad y frecuencia de las comunicaciones imponen severas restricciones que han originado importantes esfuerzos de investigación en el el área de la Robótica.

El primer uso de manipuladores teleoperados tuvo lugar en 1967 por la Unión Sovietica, los cuales recogieron muestras de la superficie lunar durante la misión del Surveyvor, existiendo un retraso de unos pocos segundos en la comunicación. Una misión de la NASA, en 1977, durante la misión del Viking, hizo la misma operación en Marte, pero en este caso el retraso en las órdenes enviadas desde la Tierra hasta la nave espacial era del orden de minutos.

Durante los años setenta fueron desarrollados los transbordadores espaciales, y con ellos surgió una nueva aplicación de los telemanipuladores. En este caso no se pretendía teleoperar desde la Tierra, sino que el propio astronauta, desde la nave, trabajará con un telemanipulador que se encuentra en el exterior. Son numerosos los retos que se plantean, tales como la falta de gravedad, el diseño de manipuladores ligeros, gran campo de acción, etc. Uno de los primeros trabajos fue llevado a cabo por la empresa canadiense SPAR para la NASA, que desarrolló un robot de grandes dimensiones, denominado sistema de manipulación remoto para transbordador (SRMS, por sus siglas en inglés, Shuttle Remote Manipulator System). Su objetivo era realizar diferentes tipos de tareas en el exterior de la nave, principalmente ensamblado de estructuras

y recogida de satélites. Otra aportación realizada por la NASA, ha sido la definición de una arquitectura propia para el control de robots teleperados, denominada NASREM.

A principios de los noventa, los japoneses comenzaron el desarrollo del Módulo Experimental Japonés (JEM, Japanese Experimental Module), como parte integrante de la estación orbital que están diseñando.

En abril de 1993, fue utilizado por primera vez un telerobot sobre el transbordador Columbia. Dicho telerobot, denominado ROTEX (Robot Technology Experiment), fue desarrollado en Alemania. Consta de numerosos sensores que le permiten trabajar de forma autónoma y manual; siendo controlado por astronautas, o bien desde tierra. El guiado del telerobot se realizó desde tierra a través de un ratón de 6 grados de libertad sin reflexión de fuerza. A su vez, al operador se le mostraba una imagen espectroscópica del entorno remoto. El sistema de control del robot se encontraba en la nave, y ejecutaba comandos relacionados a la tarea a realizar. Dichos comandos hacían uso de la información suministrada por los sensores, lo cual le permitía adaptarse al entorno de trabajo. El retraso de las comunicaciones en las pruebas realizadas fue en torno a 6 segundos, realizándose con éxito.

En los últimos años, el resurgimiento de la carrera espacial ha aumentado el número de sistemas teleoperados para el espacio. Como ejemplo de esto, puede citarse el sistema que está siendo desarrollado para su uso en la Estación Espacial Internacional (ISS, por sus siglas en inglés) por SPAR en Canadá denominado Mobile Servicing System. Este sistema se compone de dos robots:

- El primero de ellos es el SSRMS (Space Station Remote Manipulator System). Se trata de un robot de gran tamaño (17 metros) y 7 grados de libertad. La simetría de su cadena cinemática está pensada para permitirle intercambiar las funciones de cada uno de sus extremos . De esta manera, mientras un extremo esta sujeto a la estructura de exterior de las ISS el otro realiza la tarea de manipulación encomendada, pudiendo intercambiarse ambas funciones. Además, el robot está dispuesto sobre un sistema de desplazamiento (MSC) que le permite desplazarse por la estructura exterior de la estación. Esta posibilidad, junto con la de intercambiar el extremo libre y el sujeto a la estructura, le dotan de una elevada movilidad.

- El segundo de los robots es el denominado SPDM (Special Purpose Dexterous Manipulator), consiste en un robot con dos brazos manipuladores de 7 grados de libertad y un alcance de 3.3 metros cada uno. Sus capacidades se ven potenciadas por la incorporación de un sistema de cambio de herramienta y el uso de sensores de fuerza-par y visión.

Ambos robots son sistemas teleoperados controlados desde la estación espacial y las funciones a las que está previsto destinarlos son, entre otras, el ensamblado y construcción de la Estación Espacial Internacional, la inspección y mantenimiento exterior de la misma, la puesta en órbita o recuperación de robots de vuelo libre o pequeñas naves y apoyo a la tripulación en operaciones extravehiculares.

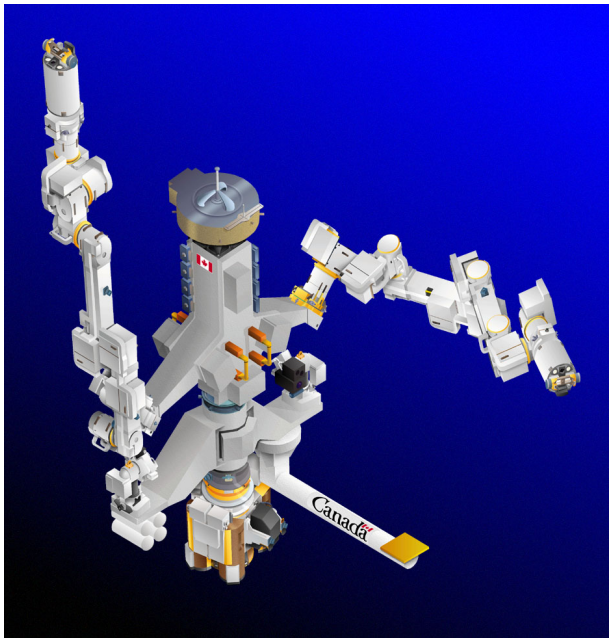


Figura 1.5. Prototipo del sistema manipulador remoto de la estación espacial (SSRMS)

1.5. Telemanipuladores en el sector médico

Son dos las principales aportaciones que puede realizar la telerobotica en el campo de la cirugía. Por una parte, la gran precisión en sus movimientos, superior a la de las manos del cirujano inevitablemente sujetas a movimientos no deseados (temblor fisiológico) o a errores de posicionamiento por falta de visibilidad, cansancio, etc. Por otra parte, la posibilidad de separar al cirujano de la sala de operaciones, pudiendo así llegar a pacientes situados en lugares alejados o en cierta medida, inaccesibles.

En definitiva, la cirugía mínimamente invasiva, si bien tiene importantes beneficios para el paciente, presenta grandes desventajas para el cirujano que ve limitada su capacidad de actuación y de sensación. Buena parte de estos inconvenientes pueden ser resueltos con ayuda de la telerobótica y de las técnicas asociadas a ella. La pérdida de la sensación de tacto motivada por el uso de largos instrumentos que pasan a través del tejido del paciente, puede ser suplida mediante la presentación al cirujano de fuerzas generadas de manera artificial a través de las medidas tomadas por unos sensores colocados en los extremos de los instrumentos. Éstos, en lugar de ser directamente manipulados por el cirujano, pueden ser teleoperados por robots manipuladores, lo que permite disminuir el cansancio del cirujano, eliminar su temblor fisiológico y aumentar la precisión haciendo uso de un adecuado escalamiento de los movimientos efectuados y los reproducidos por el robot.

La telerobótica en el sector médico fue desarrollada con el apoyo de el Departamento de Defensa de los Estados Unidos para permitir a los cirujanos, en locaciones remotas, operar a soldados heridos en el campo de batalla. El noventa por ciento de todas las muertes en combate ocurren antes de que los soldados reciban

atención médica. Por estas razones el objetivo de este sector fue el de permitir que los cirujanos atiendan inmediatamente las heridas de gravedad antes de que éstos puedan ser evacuados a hospitales militares. En este escenario, los soldados heridos serían llevados a un vehículo especial para cirugía [28].

El primer robot aprobado por la FDA para uso clínico en cirugías de abdomen, fue el sistema automatizado para endoscopias con posición óptima (AESOP, por sus siglas en inglés), en 1994, por el centro de investigación Computer Motion. Esta compañía fue fundada por una beca de investigación de la NASA, que tenia la misión de desarrollar un brazo robótico para el programa espacial. Este brazo fue después modificado para sostener un laparoscopio y reemplazar el que un asistente en la sala de operaciones sostiene. En resumen, el AESOP es un robot controlado por voz, que sostiene una cámara de vídeo durante un procedimiento de laparoscopia. El brazo mantiene estable la cámara y garantiza un correcto alineamiento en la imagen de vídeo. Esto permite que un solo cirujano, mediante su voz, realice la intervención sin la necesidad de asistentes, solo con la ayuda del AESOP.

La empresa Computer Motion, la creadora del robot AESOP, desarrolló tiempo después el telerobot Zeus. Tomó como base al robot AESOP para desarrollar un robot capaz de realizar cirugías a distancia. En este sistema, el robot controlado por voz AESOP continua con la tarea de sostener la cámara, mientras que dos manipuladores AESOP han sido modificados para sostener instrumentos quirúrgicos, que actúan como las manos derecha e izquierda del cirujano. Estos tres elementos están sujetos independientemente a una mesa de operaciones; esto permite que la mesa pueda ser colocada en diferentes posiciones sin separar los manipuladores del paciente. La consola de cirugía de Zeus se divide en dos partes, un monitor



Figura 1.6. Robot Zeus para cirugías cardíacas

proyecta una imagen tridimensional de la parte operada que puede ser vista a través de lentes con filtros polarizados. La interfaz de los manipuladores maestros transmite el movimiento de las manos del cirujano en movimientos de los instrumentos del manipulador. La segunda parte es el lugar donde el cirujano se

sienta confortablemente en una silla dentro de la consola de Zeus.

Como otro ejemplo de un sistema para cirugía teleoperada, puede citarse el sistema DaVinci, desarrollado por Intuitive Surgical Inc. El robot Da Vinci consiste en tres partes separadas. El cirujano se coloca en una posición ergonómica y ventajosa dentro de una estación de trabajo. Sus manos se adaptan a unos dispositivos maestros que actúan como una interfaz con la computadora. La computadora y un sistema de imágenes en 3D ocupan el resto del espacio en la estación de trabajo. Una torre mantiene todo el equipo electrónico de vídeo necesario para la calidad de las imágenes. Finalmente, el robot tiene tres brazos manipuladores. El brazo central es el encargado de sostener la cámara y los otros dos, los instrumentos quirúrgicos que reproducen los movimientos del cirujano. Los instrumentos quirúrgicos se mueven como una muñeca, éstos tienen 7 grados de libertad y 2 grados de libertad de rotaciones adicionales. El robot se posiciona sobre una mesa de operaciones. En 1999 se reportó la primera intervención quirúrgica exitosa con Da Vinci en un bypass de una arteria coronaria.

La telerobótica debe ser considerada como una tecnología emergente que aun está en etapa de desarrollo. Por otro lado, se han identificado dilemas éticos y legales, debido a que las cirugías a distancia interfieren en la tradicional relación de medico-paciente.



Figura 1.7. Robot DaVinci para cirugías cardíacas

Capítulo 2

Cinemática del robot industrial

La serie M-16iB/T de FANUC Robotics es una familia de robots de seis juntas montados sobre un riel. Estos robots están diseñados para brindar alta precisión, operación a alta velocidad, una plataforma amigable al usuario y confiabilidad máxima. El manipulador M-16iB 20T consta de una junta prismática y cinco juntas de revoluta accionadas por servomotores, donde sus características principales son listadas en la Tabla 2.1. Los manipuladores mecánicos o robots, están compuestos de eslabones conectados por juntas para formar

Tabla 2.1. Características del Fanuc M16iB-20T

Característica		Valor	Unidades
Cantidad de ejes		6	-
Capacidad de carga		20	kg
Repetibilidad		$\pm$0.1	mm
Rango de movimiento	θ_1	3600	mm
	θ_2	300	grados
	θ_3	460	grados
	θ_4	400	grados
	θ_5	280	grados
	θ_6	900	grados
Velocidad de movimiento máxima	θ_1	2750	mm/s
	θ_2	165	grados/s
	θ_3	175	grados/s
	θ_4	350	grados/s
	θ_5	340	grados/s
	θ_6	520	grados/s

una cadena cinemática abierta. Las juntas con las que cuenta el robot, en nuestro caso, son de rotación y prismáticas. El robot utilizado cuenta con 6 grados de libertad, que son mostrados en la Figura 2.1. En esta figura también se describe el movimiento que realiza cada uno de los ejes, donde los tres primeros ejes sirven en general para posicionamiento y los últimos tres, que forman la muñeca, para orientación. La muñeca tiene

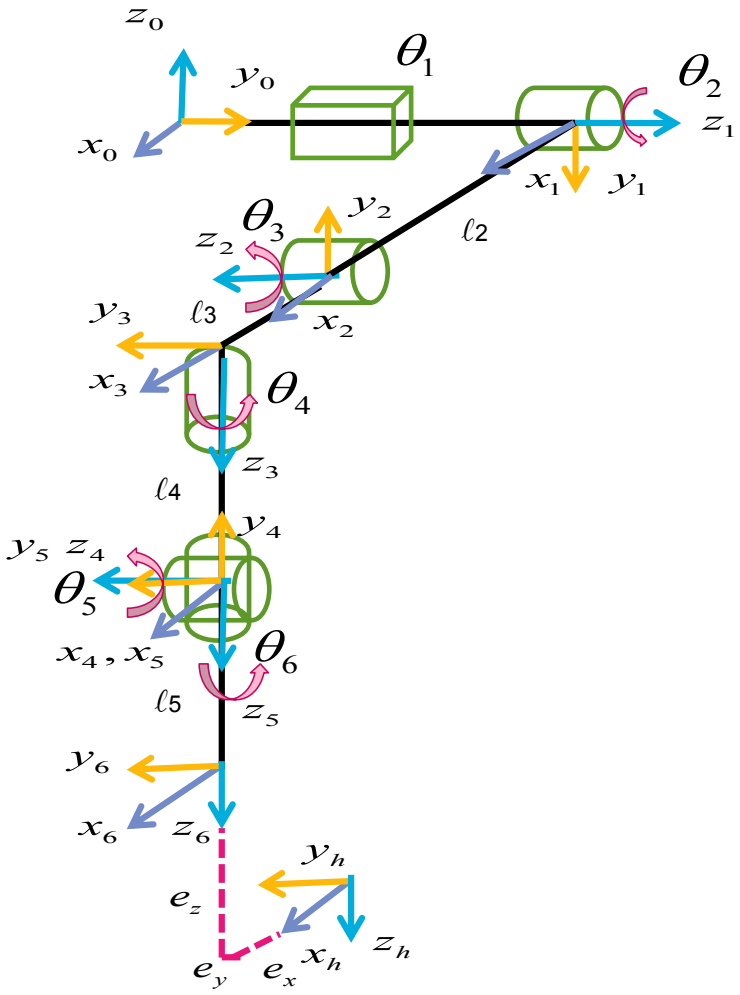


Figura 2.1. Esquema para la cinemática directa

una configuración esférica, esto es, sus ejes se intersectan en un punto común.

2.1. Modelo Cinemático Directo FANUC M16iB-20T

El modelo cinemático directo de un robot describe la relación entre la posición articular **q**, y la posición cartesiana **u** del dispositivo terminal del robot manipulador. En otras palabras, el modelo cinemático directo de un robot manipulador se define como:

$$u = f(\mathbf{q}) \tag{2.1}$$

en donde $\mathbf{q} \in \Re^n$ es el vector de juntas que se define como $\mathbf{q} = [\theta_1, \theta_2, ..., \theta_n]^T$, $\mathbf{u} \in \Re^m$ es el vector de coordenadas operacionales que describe la posición y orientación del efector final, f es una función vectorial que realiza el mapeo de $f : \Re^n \longmapsto \Re^m$, n es el número de grados de libertad del manipulador

y m el número de variables requeridas para la tarea de interés. Para realizar este análisis, se requiere de la designación de un marco de referencia o sistema coordenado unido a cada eslabón del mismo y de una referencia fija que generalmente está situada en la base del robot.

Tabla 2.2. Dimensiones del robot Fanuc M16iB-20T

Segmento	Dimensiones (mm)
ℓ_2	770
ℓ_3	100
ℓ_4	740
ℓ_5	100

Se emplea el concepto de transformación homogénea para describir, con respecto a un sistema de referencia fijo, la posición y orientación de los eslabones que componen el brazo del robot. La Figura 2.1 muestra los elementos necesarios para obtener el modelo cinemático del robot utilizado en los experimentos. Cada cilindro representa a una junta de revoluta, es decir, una rotación alrededor de un eje asociado a la junta. El prisma representa un movimiento lineal sobre el eje asociado, como se aprecia en la primera junta de nuestro manipulador. (Ver Figura 2.1)

La relación matricial (2.2) es la correspondiente a la rotación alrededor del eje x.

$$\mathbf{R}_{(x,\theta)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.2}$$

La relación matricial (2.3) es la que corresponde a una rotación alrededor del eje y.

$$\mathbf{R}_{(y,\theta)} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

La relación matricial (2.4) es la correspondiente a una rotación alrededor del eje z.

$$\mathbf{R}_{(z,\theta)} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

La relación matricial (2.5) corresponde a una traslación en los ejes x,y y z en una distancia a, b y c

respectivamente.

$$\mathbf{T}_{(a,b,c)} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

La transformación homogénea $\mathbf{A}_i$ es aquella que relaciona al sistema coordenado i con el sistema coordenado $i-1$. Para el caso de la primera junta, esta relación se expresa en la ecuación (2.6) y significa una traslación a lo largo del eje y un desplazamiento d_1.

$$\mathbf{A}_1 = \mathbf{T}_{(y,d_1)}\mathbf{R}_{(x,90^\circ)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

$$\mathbf{A}_2 = \mathbf{T}_{(x,\ell_2\cos(\theta_2))}\mathbf{T}_{(y,\ell_2\cos(\theta_2))}\mathbf{R}_{(x,180^\circ)} = \begin{bmatrix} 1 & 0 & 0 & \ell_2\cos(\theta_2) \\ 0 & -1 & 0 & \ell_2\sin(\theta_2) \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.7}$$

$$\mathbf{A}_3 = \mathbf{R}_{(z,\theta_3)}\mathbf{T}_{(x,\ell_3)}\mathbf{R}_{(x,90^\circ)} = \begin{bmatrix} \cos(\theta_3) & 0 & \sin(\theta_3) & \ell_3\cos(\theta_3) \\ \sin(\theta_3) & 0 & -\cos(\theta_3) & \ell_3\sin(\theta_3) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

$$\mathbf{A}_4 = \mathbf{R}_{(z,-\theta_4)}\mathbf{T}_{(z,\ell_4)}\mathbf{R}_{(x,-90^\circ)} = \begin{bmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) & 0 \\ -\sin(\theta_4) & 0 & \cos(\theta_4) & 0 \\ 0 & -1 & 0 & \ell_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

$$\mathbf{A}_5 = \mathbf{R}_{(z,\theta_5)}\mathbf{R}_{(x,90^\circ)} = \begin{bmatrix} \cos(\theta_5) & 0 & \sin(\theta_5) & 0 \\ \sin(\theta_5) & 0 & -\cos(\theta_5) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.10}$$

$$\mathbf{A}_6 = \mathbf{R}_{(z,-\theta_6)}\mathbf{T}_{(z,\ell_5)} = \begin{bmatrix} \cos(\theta_6) & \sin(\theta_6) & 0 & 0 \\ -\sin(\theta_6) & \cos(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & \ell_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.11}$$

La matrices $\mathbf{A}_1 ... \mathbf{A}_6$, donde las dimensiones $\ell_2, \ell_3, \ell_4, \ell_5$ se presentan en la Tabla 2.2, describen la posición-orientación del extremo libre del robot referido al sistema coordenado en la base del manipulador (x_0, y_0, z_0). En la ecuación (2.12) se realiza el producto de estas matrices.

$$\mathbf{A} = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6 \tag{2.12}$$

Con el resultado del producto matricial anterior se obtiene la ecuación cinemática del robot sin herramienta. El siguiente paso es obtener la matriz constante que describe la posición y orientación del extremo de la herramienta con referencia al extremo libre del robot. La matriz que describe esta transformación se denomina **H**, y se puede definir en términos generales como:

$$\mathbf{H} = \mathbf{T}_{(x,-e_x)}\mathbf{T}_{(y,-e_y)}\mathbf{T}_{(z,e_z)} \tag{2.13}$$

La cantidades e_x,e_y,e_z se especifican en la Tabla 2.3.

Tabla 2.3. Dimensiones de la herramienta

Segmento	Dimensiones (mm)
e_x	223
e_y	3.4
e_z	388.2

La matriz homogénea final que transforma las coordenadas de un punto desde el marco de referencia asociado a la herramienta, al sistema coordenado de referencia colocado en la base del robot (x_0, y_0, z_0), se puede obtener realizando el producto matricial de las ecuaciones (2.12) y (2.13).

$$\mathbf{T} = \mathbf{A}_1\mathbf{A}_2\mathbf{A}_3\mathbf{A}_4\mathbf{A}_5\mathbf{A}_6\mathbf{H} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & x \\ r_{21} & r_{22} & r_{23} & y \\ r_{31} & r_{32} & r_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.14}$$

2.2. Modelo Cinemático Inverso FANUC M16iB-20T

Esta sección es incluida debido a que, dentro de nuestro esquema de teleoperación, el dispositivo maestro envía las posiciones al robot manipulador en variables cartesianas, ya que éstas son más intuitivas para el operador. Sin embargo, para un movimiento óptimo del robot, es necesario conocer el valor de las variables de junta para evaluar un índice de desempeño cinemático. Una solución a este problema es la utilización de la metodología de cinemática inversa [23].

El problema de la cinemática inversa es el de encontrar las variables de juntas en términos de la posición y orientación del extremo del manipulador, es decir, es la relación entre el espacio cartesiano **u** y la posición articular **q**. Este mapeo se define como:

$$\mathbf{q} = f^{-1}(\mathbf{u}) \tag{2.15}$$

Aunque este problema es en general complicado sucede que, para el manipulador utilizado que cuenta con seis articulaciones, con las tres últimas intersectándose en un mismo punto (es decir, con una muñeca de configuración esférica), es posible desacoplar el problema de cinemática inversa en dos problemas más sencillos, los cuales se conocen como cinemática de posición inversa y cinemática de orientación inversa. El primero de ellos consiste en encontrar la posición del punto de intersección de los ejes de la muñeca, llamado centro de la muñeca, y entonces encontrar la orientación de la misma.

El punto de partida para la resolución de la cinemática inversa por medio del método de desacoplamiento cinemático, es que el movimiento de los últimos tres eslabones alrededor de sus ejes no cambia la posición del centro de la muñeca esférica. En este caso, el centro de la muñeca se localiza simplemente con una traslación del valor ℓ_5, entonces la posición del centro de la muñeca, con ayuda del modelo cinemático representado en la ecuación (2.14), estará dado por la siguiente expresión:

$$x_w = x - \ell_5 r_{13} \tag{2.16}$$

$$y_w = y - \ell_5 r_{23} \tag{2.17}$$

$$z_w = z - \ell_5 r_{33} \tag{2.18}$$

donde los valores de r_{13}, r_{23}, r_{33}, se obtienen de la ecuación (2.14).

La posición del centro de la muñeca depende sólo de las tres primeras variables de junta. Con ayuda de la figura 2.2, se pueden calcular las tres primeras variables de junta de forma geométrica. Recordando que la primera junta del manipulador es prismática, el problema del cálculo para la primera junta se simplifica y se tiene que:

$$\theta_1 = y_w \tag{2.19}$$

Para el cálculo de θ_2 y θ_3 se hace uso de la figura 2.2 y se obtiene:

$$\theta_2 = \alpha + \beta \tag{2.20}$$

$$\theta_3 = \gamma - \delta \tag{2.21}$$

considerando que

$$\alpha = \tan^{-1}\left(\frac{z_w}{x_w}\right) \Rightarrow \alpha = \operatorname{atan2}(z_w, x_w) \tag{2.22}$$

$$\cos\beta = \frac{-P^2 + R^2 + {l_1}^2}{2Rl_1} = D \tag{2.23}$$

$$\beta = \operatorname{atan2}(\sqrt{1 - D^2}, D) \tag{2.24}$$

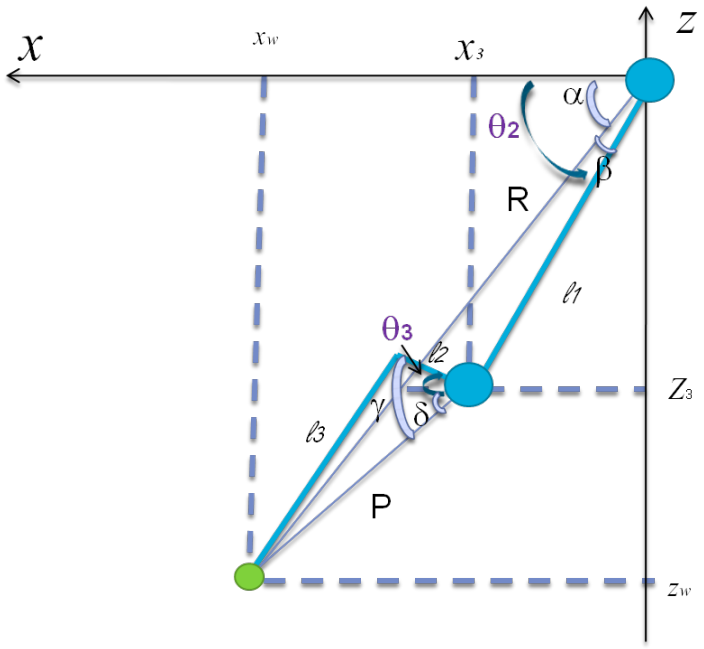


Figura 2.2. Esquema para el cálculo de la cinemática inversa

$$\gamma = \tan^{-1}\left(\frac{l_3}{l_2}\right) \Rightarrow \gamma = \text{atan2}(l_3, l_2) \tag{2.25}$$

$$\delta = \tan^{-1}\left(\frac{z_w - z_3}{x_w - x_3}\right) \Rightarrow \delta = \text{atan2}(z_w - z_3, x_w - x_3) \tag{2.26}$$

donde

$$R = \sqrt{{x_w}^2 + {z_w}^2}; P = \sqrt{{\ell_3}^2 + {\ell_2}^2}$$

La función trigonométrica $\theta = atan2(y, x)$, calcula la función tangente inversa, donde x y y son el coseno y seno, respectivamente, del angulo θ. Esta función utiliza el signo de x y y, para seleccionar el cuadrante apropiado para el ángulo θ. Específicamente, $atan2(x, y)$ esta definida para todo $(x, y) \neq (0, 0)$ y es igual al único ángulo, el cual cumpla: $sen\theta = \frac{y}{(x^2+y^2)^{\frac{1}{2}}}$, $cos\theta = \frac{x}{(x^2+y^2)^{\frac{1}{2}}}$.

Usando las ecuaciones (2.19) a (2.21), se obtiene la matriz de rotación $\mathbf{R}_3^0$, la cual depende solamente de esos primeros tres valores. Con esta matriz de rotación se puede determinar la orientación del efector final de la expresión (2.28).

$$\mathbf{R} = \mathbf{R}_3^0 \mathbf{R}_6^3 \tag{2.27}$$

$$\mathbf{R}_6^3 = \left(\mathbf{R}_3^0\right)^{-1} \mathbf{R}_6^0 = \left(\mathbf{R}_3^0\right)^T \mathbf{R} \tag{2.28}$$

Por lo tanto, la orientación se obtuvo de manera algebraica de tal forma que:

$$\theta_4 = -\text{atan2}(R_{23}, R_{13}) \tag{2.29}$$

$$\theta_5 = \text{atan2}(\sqrt{1 - {R_{33}}^2}, R_{33}) \tag{2.30}$$

$$\theta_6 = -\text{atan2}(R_{32}, R_{31}) \tag{2.31}$$

donde R_{ij} representa al término en el iésimo renglón y la j-ésima columna de la matriz $\mathbf{R}_6^3$.

2.3. Jacobiano del Manipulador

Matemáticamente, las ecuaciones de cinemática directa definen una función entre el espacio de posiciones y orientaciones cartesianas, y el espacio de posiciones de las juntas. Las relaciones de velocidad se determinan entonces por medio del Jacobiano de esta función [22]. Este Jacobiano es una función matricial que puede asociarse con la versión vectorial de la derivada ordinaria de una función. El Jacobiano, o matriz Jacobiana, es una de las cantidades más importantes en el análisis y control del movimiento del robot, ya que permite establecer varios aspectos de manipulación robótica como:

- Planeación y ejecución de trayectorias suaves.
- Determinación de configuraciones singulares.
- Ejecución de movimiento coordinado tipo antropomórfico.
- Derivación de ecuaciones de movimiento dinámicas.
- Determinación del índice de manipulabilidad de un robot.
- Transformación de fuerzas y torques ejercidos en el efector final a los torques requeridos en las juntas del manipulador.

Para un manipulador de n grados de libertad, se deriva primero al Jacobiano que representa la transformación instantánea entre un vector de velocidades de junta, con n elementos y el vector de 6 elementos que contiene a las componentes de velocidad lineal y angular del efector final. El Jacobiano es entonces una matriz de 6 x n. El mismo enfoque se usa para determinar la transformación entre las velocidades de junta y la velocidad lineal y angular de cualquier punto en el manipulador. En lo que respecta a las configuraciones singulares, se trata de configuraciones en las que el robot pierde uno o más grados de libertad.

Como se vio en la sección 2.1, la matriz de transformación (2.14) desde el marco de referencia en el efector final al marco de referencia de la base, se puede reescribir de la siguiente forma:

$$\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{d} \\ 0 & 1 \end{bmatrix} \tag{2.32}$$

donde $\mathbf{d} = [\ x \quad y \quad z\ ]^T$, describe la posición final del manipulador con respecto a su sistema coordenado en la base y $\mathbf{R}$ se define en la ecuación (2.27).

Al moverse el robot, las variables de junta, así como la posición y orientación de este efector, cambiarán en función del tiempo. El objetivo de la matriz Jacobiana será relacionar la velocidad lineal y angular de este elemento, con el vector de velocidades de junta $\dot{\mathbf{q}}$. Sea $\vec{\omega}$ la velocidad angular del efector y $\mathbf{v} = \dot{\mathbf{d}}$ la velocidad lineal, se buscan expresiones de la forma:

$$\mathbf{v} = \mathbf{J}_v \dot{\mathbf{q}} \tag{2.33}$$

$$\vec{\omega} = \mathbf{J}_w \dot{\mathbf{q}} \tag{2.34}$$

donde $\mathbf{J}_w$ y $\mathbf{J}_v$ son matrices de 3 x n. De manera alternativa puede escribirse,

$$\begin{bmatrix} \mathbf{v} \\ \vec{\omega} \end{bmatrix} = \mathbf{J}\dot{\mathbf{q}} \tag{2.35}$$

donde,

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_v \\ \mathbf{J}_w \end{bmatrix} \tag{2.36}$$

La matriz $\mathbf{J}$ se denomina Jacobiano del manipulador o simplemente Jacobiano y sus dimensiones son de 6 x n, donde n es el número de articulaciones del robot. A continuación se describen medios para evaluar esta matriz.

2.3.1. Velocidad Angular

Las velocidades angulares pueden ser sumadas vectorialmente, considerando que se expresan respecto a un marco de referencia común. Es posible, entonces, determinar la velocidad angular del efector final, relativa a la base, expresando la velocidad angular de cada articulación respecto al sistema en la base, y sumándolas. Si $\hat{\lambda}_{i-1}$ representa al vector unitario a lo largo del eje de la junta que gira a una velocidad angular $\dot{\theta}_i$, la cual es cero en el caso de una junta prismática, entonces

$$\vec{w}_{i-1/i} = \dot{\theta}_i \hat{\lambda}_{i-1} \tag{2.37}$$

$$\vec{w}_{0/n} = \dot{\theta}_1 \hat{\lambda}_0 + \dot{\theta}_2 \hat{\lambda}_1 + ... + \dot{\theta}_n \hat{\lambda}_{n-1} \tag{2.38}$$

Ahora, si la *iésima* junta es de revoluta , entonces la variable de junta q_i corresponde a θ_i y el eje de rotación es z_{i-1}. Así, la velocidad angular de la *iésima* articulación expresada respecto al marco de referencia anterior $(i-1)$ está dada por,

$$\vec{w}_{i-1/i} = \dot{q}_i \mathbf{k} \tag{2.39}$$

donde $\mathbf{k} = [0, 0, 1]^T$. Si la *iésima* junta es prismática, entonces la orientación del marco i respecto al marco $i-1$ no cambia y $\vec{w}_{i-1/i} = 0$. Así, si la junta es prismática, la velocidad angular del efector no depende de q_i que es, en este caso, d_1. Entonces, la velocidad angular del efector final $\vec{w}_{0/n}$, respecto al sistema de la base es,

$$\vec{w}_{0/n} = \rho_1 \dot{q}_1 \mathbf{k} + \rho_2 \dot{q}_2 R_{0/1}\mathbf{k} + ... + \rho_n \dot{q}_n \mathbf{R}_{0/n-1}\mathbf{k} = \sum_{i=1}^{n} \rho_1 \dot{q}_1 \mathbf{z}_{i-1} \tag{2.40}$$

donde

$$\mathbf{z}_{i-1} = \mathbf{R}_{0/i-1}\mathbf{k} \tag{2.41}$$

denota al vector unitario a lo largo del eje de rotación del marco $i-1$, expresado con respecto al sistema coordenado en la base; note que $\mathbf{z}_0 = \mathbf{k} = [0, 0, 1]^T$. Además, ρ_i es igual a 1 si la junta es de revoluta y a 0 si es prismática. Bajo esta nomenclatura, el Jacobiano para velocidades angulares es,

$$\mathbf{J}_w = [\rho_1 \mathbf{z}_0, ..., \rho_n \mathbf{z}_{n-1}] \tag{2.42}$$

2.3.2. Velocidad Lineal

La velocidad lineal del efector final es simplemente $\dot{\mathbf{d}}$. Aplicando la regla de la cadena de diferenciación se tiene,

$$\dot{\mathbf{d}} = \sum_{i=1}^{n} \frac{\partial \mathbf{d}}{\partial q_i} \dot{q}_i \tag{2.43}$$

Es claro que la *iésima* columna de $\mathbf{J}_v$ es simplemente:

$$\mathbf{J}_{vi} = \frac{\partial \mathbf{d}}{\partial q_i} \tag{2.44}$$

Mas aún, esta expresión es la velocidad lineal del efector final que resultaría si $\dot{q}_i$ fuera igual a 1 y el resto de las velocidades de junta $\dot{q}_j$ fueran cero. En otras palabras, la *iésima* columna del Jacobiano puede generarse al mantener todas las juntas fijas excepto la *iésima*, la cual se desplaza a velocidad unitaria.

Entonces la matriz Jacobiana $\mathbf{J}_v$ está dada por:

$$\mathbf{J}_v = [\mathbf{J}_{v1}, ..., \mathbf{J}_{vn}] \tag{2.45}$$

2.4. Manipulabilidad

La determinación de un mecanismo y tamaño del robot manipulador en la etapa de diseño y la determinación de la postura del manipulador en el espacio de trabajo, para el desempeño de una tarea establecida en la etapa de operación, es un problema que ha sido abordado con base en la experiencia e intuición de

los desarrolladores. Una de varias características tomadas en cuenta para resolver este problema, es la facilidad de cambiar arbitrariamente la posición y orientación del efector final del manipulador. Entonces, es beneficioso para el diseño, control y planeación de tareas o trayectorias del robot manipulador, la existencia de una medida cuantitativa de la habilidad de manipulación del brazo del robot, dependiendo de la posición y orientación del efector final.

Generalmente, en los robots manipuladores, existen determinadas configuraciones que hacen que su matriz Jacobiana no tenga rango completo. Tales configuraciones son llamadas singulares y, como se sabe, en estos casos no es posible resolver el modelo inverso de velocidad del robot. En consecuencia, una configuración singular incapacita localmente al manipulador para desplazar el órgano terminal en una cierta dirección dentro del espacio cartesiano, lo cual le resta manipulabilidad.

La posibilidad de que un manipulador asuma configuraciones singulares durante la ejecución de una tarea, fue planteada en uno de los primeros trabajos que trataron el problema del modelado cinemático de los manipuladores. Varios autores se han dedicado a la caracterización de las configuraciones singulares de manipuladores; otros consideran la prevención de configuraciones singulares y la facilidad de inversión de la matriz Jacobiana como criterios para guiar el diseño de manipuladores [16], [17], [18].

2.4.1. Índice de Manipulabilidad

Tsuneo Yoshikawa [15], en 1983, usó la siguiente medida cuantitativa de manipulabilidad para manipuladores redundantes, es decir, cuando el Jacobiano no es una matriz cuadrada.

Definición: El índice de manipulabilidad es una magnitud escalar dada por:

$$j = \sqrt{\det(\mathbf{J}(\mathbf{q})\mathbf{J}^T(\mathbf{q}))} \tag{2.46}$$

En nuestro caso, el manipulador tiene 6 grados libertad, entonces $n = 6$, y se tiene una matriz cuadrada de dimensión 6×6, por estas razones se considera un manipulador no redundante. El índice de manipulabilidad se reduce a la siguiente ecuación:

$$j = |\det \mathbf{J}(\mathbf{q})| \tag{2.47}$$

El valor de j puede llegar a ser 0 si solo si $rank\mathbf{J}(q) \leq m$, esto es, cuando el manipulador se encuentra en una posición singular.

Las singularidades de un robot son configuraciones no deseadas, ya que pueden significar una operación excesiva de los servomotores, o la imposibilidad de realizar localmente algunos movimientos, o una baja

precisión en el posicionamiento del efector final, es decir, la manipulabilidad del mismo es seriamente deteriorada. Desde este punto de vista, se puede considerar el índice de manipulabilidad como una medida de qué tanto se aleja un manipulador de puntos singulares.

2.4.2. Mecanismo de 2 grados de libertad

Para explicar de manera un poco más clara el concepto de índice de manipulabilidad, se incluye esta subsección. Considere un manipulador planar de 2 grados de libertad (θ_1, θ_2), con l_1 y l_2 como las longitudes de sus eslabones como se muestra en la Figura (2.3), el cual, es el caso más simple de manipuladores multivariables.

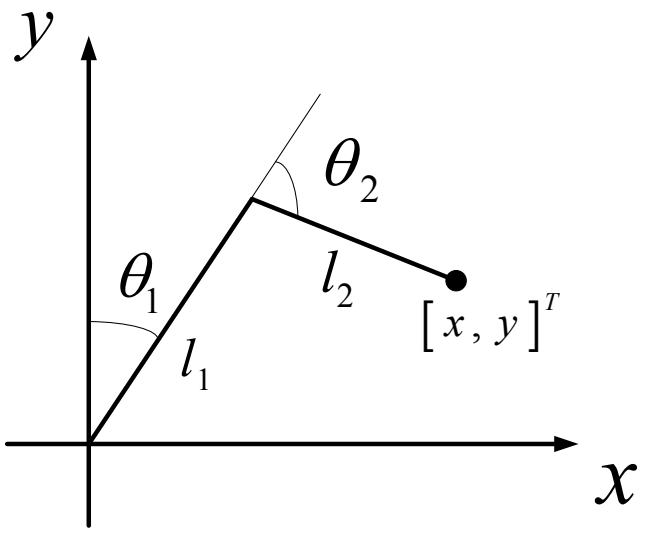


Figura 2.3. Mecanismo planar de 2 grados de libertad

Cuando el extremo del manipulador tiene la posición $[x, y]^T$, éste es considerado como nuestro vector u, la matriz Jacobiana del manipulador está dada por [14]:

$$\mathbf{J} = \begin{bmatrix} l_1\cos(\theta_1) + l_2\cos(\theta_1+\theta_2) & l_2\cos(\theta_1+\theta_2) \\ l_2\sin(\theta_1) - l_2\sin(\theta_1+\theta_2) & -l_2\sin(\theta_1+\theta_2) \end{bmatrix} \tag{2.48}$$

Entonces, el índice de manipulabilidad j es;

$$j = |\det \mathbf{J}| = l_1 l_2 \,|\sin\theta_2| \tag{2.49}$$

Se puede ver, de la ecuación (2.49), que el manipulador alcanza la postura con el índice de manipulabilidad más elevado cuando $\theta_2 = \pm 90°$, con cualquier valor dado de l_1, l_2 y θ_1. Se enfoca en el término $l_1 l_2$ y lo

igualamos a un valor máximo P, se tiene que;

$$l_1 l_2 = P \tag{2.50}$$

bajo la condición de que

$$l_1 + l_2 = k \tag{2.51}$$

donde k es el valor de una longitud total constante. Despejando la longitud l_2 se tiene que:

$$l_2 = k - l_1 \tag{2.52}$$

sustituyendo (2.52) en (2.50)

$$P = l_1(k - l_1) = kl_1 - {l_1}^2 \tag{2.53}$$

Para obtener el máximo valor de P en la ecuación (2.53), se deriva con respecto a la variable l_1, y se iguala a cero, se tiene:

$$k - 2l_1 = 0 \tag{2.54}$$

Despejando la variable l_1 se obtiene:

$$l_1 = \frac{k}{2} \tag{2.55}$$

Despejando l_2 de (2.51) y sustiyendo el valor de l_1 se tiene:

$$l_2 = \frac{k}{2} \tag{2.56}$$

Entonces el índice de manipulabilidad alcanza su máximo valor cuando $l_1 = l_2$.

Cuando un brazo humano es considerado como un manipulador de dos grados de libertad, esto es, dejando de lado los grados de libertad que contienen el hombro y la muñeca, se satisface aproximadamente la relación l_1=l_2. Por otra parte, cuando se sujeta cualquier objeto con nuestras manos o se intenta escribir, el ángulo de codo se encuentra alrededor del valor de 90° para una manipulabilidad máxima. Por lo tanto, se puede decir que las personas al tomar un objeto o escribir, inconscientemente asumen la mejor postura del brazo desde el punto de vista de la manipulabilidad.

Capítulo 3

Características del dispositivo móvil

En este capítulo se analizan los principales componentes para el diseño de nuestra interfaz. Debido a que el dispositivo seleccionado para llevar a acabo la tarea de dispositivo maestro es el iPod Touch, es necesario conocer todas las herramientas que se tengan disponibles para cumplir satisfactoriamente los objetivos planteados, entre ellos, el diseño de una interfaz ergonómica e intuitiva.

Los dispositivos móviles de Apple (Ver Figura 3.1), al igual que los denominados Smart Phones, son capaces de realizar importantes procesamientos de cómputo y combinar una amplia gama de elementos tecnológicos que los hace únicos en la actualidad. Poseen las características de un reproductor de música y video, una consola pórtatil de videojuegos y son capaces de comunicarse usando los protocolos de comunicación Bluetooth y WiFi. También ofrecen servicios como la capacidad de leer documentos, comunicación 3G y acceso a Internet.



Figura 3.1. Dispositivos móviles de Apple

Apple Inc. desarrolló el sistema operativo iOS [19], anteriormente denominado iPhone OS, para los dispositivos móviles iPod Touch, iPhone y iPad. El iOS está basado en una variante del Mach Kernel DE

Mac OS X, el cual permite el desarrollo de las aplicaciones denominadas Apps para estos dispositivos móviles, con una gama de herramientas basadas en el lenguaje Objetive-C. El sistema operativo iOS tiene 4 capas de abstracción: la capa del núcleo del Sistema Operativo (SO), la capa de Servicios Principales, la capa de Medios de Comunicación y la capa Cocoa Touch.

La interfaz de usuario de iOS se basa en el concepto de manipulación Multi-Touch, es decir, mediante gestos de contacto con su pantalla táctil. Los elementos de la interfaz se componen por deslizadores, interruptores y botones. La respuesta es inmediata, lo que se traduce en una interfaz fluida. La interacción con el SO se realiza mediante gestos como deslizar, pellizcar y presionar. Algunas aplicaciones utilizan los acelerómetros internos de los dispositivos para responder a movimientos y gestos como sacudir el aparato o rotarlo. A partir de la versión 4 del iOS, los dispositivos móviles soportan el uso de 7 interfaces de aplicación para programas multitareas.

3.1. Desarrollo de aplicaciones para el Ipod Touch

Apple Inc. proporciona una herramienta SDK (Software Development Kit), como ambiente de desarrollo de software para los dispositivos de Apple que utilizan iOS. Este ambiente permite el desarrollo de aplicaciones para iPhone, iPod Touch e iPad a cualquiera de sus cuatro niveles de abstracción, desde la capa del núcleo del sistema operativo hasta la capa Cocoa Touch. Para poder desarrollar aplicaciones, es necesario seguir una serie de pasos listados a continuación:

- Unirse al programa de Apple Developer Center, el cuál tiene un costo de 99 dólares, y permite descargar las aplicaciones diseñadas a un solo dispositivo.
- Contar con un iPhone, iPod Touch o iPad.
- Contar con una computadora Mac, con un sistema operativo Leopard o Snow instalado.
- Firmar un contrato de confidencialidad, que evite la difusión del programa creado al público en general.
- Descargar la última versión del SDK.

Para verificar que las nuevas aplicaciones han sido realizadas por desarrolladores adscritos, Apple se encarga de entregar un certificado que es necesario para realizar cualquier prueba en el dispositivo. Una vez terminado el desarrollo de una aplicación, ésta puede distribuirse en una de las tres diferentes formas previstas por Apple Inc.:

i) A través de la Store de Apple.
ii) Mediante una empresa particular que la distribuye a sus empleados.
iii) Sobre una red Ad-hoc de hasta 100 iPhones.

Los desarrolladores son libres de asignar a sus aplicaciones el precio deseado; Apple podrá distribuir estas aplicaciones, de acuerdo al tipo de membresía que tenga el desarrollador, de modo que éste obtiene el setenta por ciento del precio fijado, mientras que Apple retiene el treinta por ciento restante. Sin embargo, los desarrolladores pueden optar por ofrecer gratuitamente las aplicaciones que desarrollen. Este modelo de distribución para el software iPhone hace imposible liberar el software basado en el código autorizado con la Licencia Pública General Versión 3 (GPLV3 pos sus siglas en ingles), la cual garantiza la libertad de compartir y alterar los programas libres.

Cualquier programa que modifique el código autorizado bajo GPLV3 también debe tener licencia. Un desarrollador no puede de distribuir una aplicación autorizada bajo el GPLV3 sin distribuir también el certificado (que Apple posee) para permitir la carga de las versiones modificadas de aquel software. El ambiente de desarrollo SDK permite crear aplicaciones para el iPod, emulando su comportamiento en una computadora Mac. Una vez que la aplicación ha sido probada mediante esta emulación, ya puede ser cargada en el dispositivo, siempre y cuando se cuente con los permisos correspondientes, tal como se ha mencionado.

Diseñar una aplicación para el Ipod Touch requiere el marco de trabajo UIkit que cuenta con las herramientas XCode e Interface Builder para la creación de ésta. El lenguaje de programación utilizado es Objetive C, el cual es un combinación del lenguaje de programación ANSI C y la programación orientada a objetos. A su vez, se utilizará la estrategia de diseño Modelo-Vista-Controlador (MVC por sus siglas en inglés).

La estrategia de diseño MVC, consta de dividir el código en áreas funcionales independientes. El lado del modelo define lo que se desea que realice la aplicación en forma de código y es responsable de mantenerlo en forma segura y de fácil acceso; la parte de vista es la interfaz que crea el usuario para la aplicación y no tiene conocimiento del origen de datos representada en ella. Finalmente, la parte de controlador actúa como un puente entre el modelo y la vista, facilitando las actualizaciones entre ellos, como se muestra en la Figura 3.2.

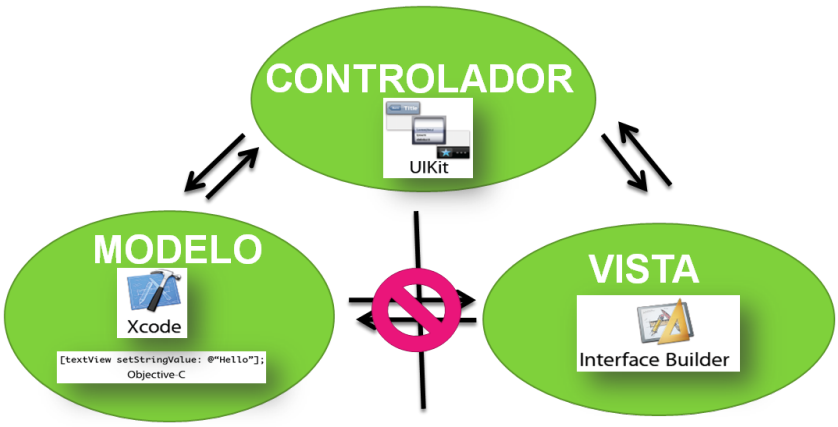


Figura 3.2. Estrategia de diseño MVC

3.2. Pantalla Táctil

Una pantalla táctil es la que permite, mediante un toque directo sobre la superficie, la entrada de datos y órdenes al dispositivo que también actúa como periférico de salida, mostrando los resultados introducidos previamente. La pantalla del iPod Touch es verdaderamente una pieza maestra de ingeniería. La pantalla táctil es uno de los factores claves en la manipulabilidad del dispositivo, debido a que la pantalla puede detectar múltiples toques y, casi al mismo tiempo, puede seguir la trayectoria de cada toque de forma independiente.

Existen diferentes tecnologías de implementación de pantallas táctiles: resistiva, onda acústica superficial, capacitivas, infrarrojos, galga extensiométrica, imagen óptica y reconocimiento de pulso acústico [21]. El Ipod Touch utiliza una pantalla táctil por capacitancia. La idea básica consiste en tomar ventaja de las propiedades eléctricas del cuerpo humano. Cuando se toca una superficie capacitiva, la cantidad de carga que tiene cambia al momento. Esta es la razón por la que el dispositivo requiere que sea tocado con los dedos, de modo que los materiales de aislamiento, como guantes, plumas y lápices, no producen los mismos cambios en el circuito capacitivo. La pantalla del dispositivo, como se ve en la Figura 3.3, consta de:

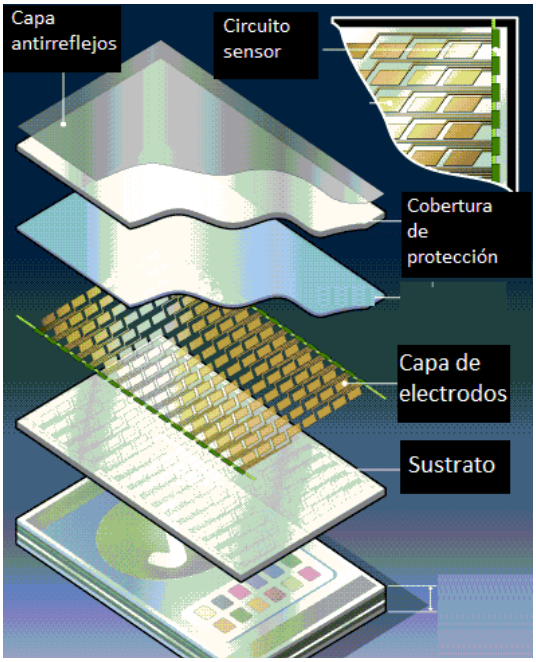


Figura 3.3. Partes que componen la pantalla del Ipod Touch

- Sustrato: Es la base de vidrio de todo el sistema sensible.
- Capa de electrodos: Es una capa transparente que se encuentra en el corazón de la pantalla multi

touch (permite varios contactos a la vez). Cuando un objeto eléctricamente cargado se acerca, cambia la carga de este electrodo.

- Circuito sensor: Interpreta el cambio de carga de los electrodos y envía la información al procesador, que determina el o los puntos de toque mediante algoritmos en sus sistemas.
- Cobertura de protección.
- Capa antirreflejos: Es la capa externa, la cual debe ser robusta, ya que recibe contacto físico del operador o "toques" y está expuesta al polvo, grasa y en general al medio ambiente.

Cuando se presiona la pantalla táctil, comienza un complejo mecanismo electrónico y matemático para determinar la posición del contacto en la pantalla, para poder activar la función asociada que debe ponerse en marcha. A grandes rasgos, el algoritmo para determinar el punto de contacto sobre la superficie, como lo explica la Figura 3.4 es:

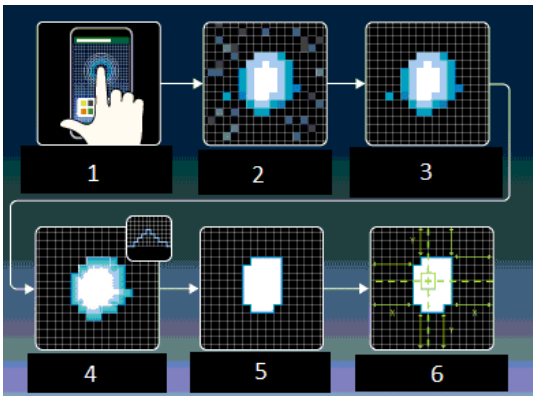


Figura 3.4. Algoritmo de reconocimiento

1. La pantalla del dispositivo registra el contacto en su superficie.
2. Se produce la captura en crudo de todos los datos producidos alrededor del contacto.
3. Las interferencias alrededor del punto de contacto son filtradas.
4. El sistema mide el grado de los puntos de presión del toque.
5. Con estos datos, el área de contacto es establecida.
6. Finalmente, son calculadas las coordenadas del punto de contacto.

Para interactuar con la pantalla táctil, el iPod reconoce varios tipos de contacto. Un contacto se define como cualquier secuencia de eventos que pasan desde el tiempo en que el usuario toca la pantalla, con uno o más dedos, hasta el momento cuando retira sus dedos de la pantalla.

3.3. Acelerómetros

Una de las características más innovadoras del iPhone y el iPod Touch es el acelerómetro interno. Los acelerómetros son herramientas modernas integradas dentro del dispositivo móvil, que permiten conocer cómo está siendo sostenido el dispositivo y si se encuentra en movimiento. El acelerómetro permite detectar la orientación del dispositivo y, de esta forma, adaptar el contenido de la pantalla a una nueva orientación.

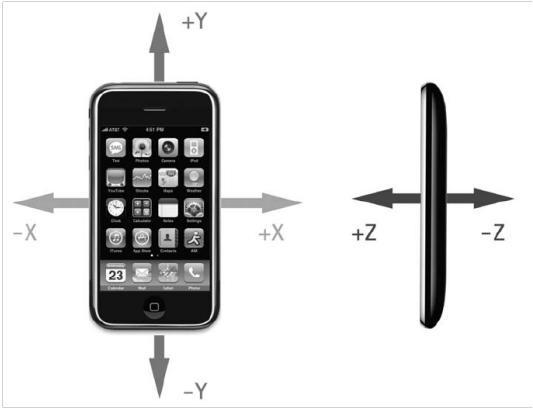


Figura 3.5. Dirección de los ejes coordenados en el Ipod Touch

El acelerómetro del iPod [20] mide aceleración y gravedad, cuantificando el aumento de fuerza inercial en una dirección determinada por el fabricante, como se muestra en la Figura 3.5. Los acelerómetros dan medidas en unidades "g" de aceleración, entonces, un valor de 1 en cierta posición significa que un g está medido en esa dirección particular, como se ilustra en la Figura 3.6. Es decir, si el aparato está siendo sostenido sin ningún movimiento, habrá aproximadamente 1 g de fuerza ejercida en él, por la aceleración de la gravedad de la Tierra. Se tiene que señalar que el valor de 1 g, cuando el dispositivo no está en movimiento, es el valor ideal, ya que en la vida real este valor cambia constantemente su valor pues estos dispositivos tienen una tasa de muestreo máxima de 10 milisegundos, pero puede variar hasta 1 segundo. Los acelerómetros usados en el Ipod dan una lectura máxima de 2.3 g cuando sienten la presencia de un movimiento brusco en alguno de sus ejes o en caída libre. Otra característica importante es que, para cada eje, el dispositivo tiene una resolución de 0.018 g.

3.4. Interfaz

Como se mencionó anteriormente, en el desarrollo de la interfaz del dispositivo maestro se deben tener en cuenta dos puntos importantes. El primero es que se debe considerar que la interfaz sea entendible por diferentes usuarios, no solo por especialistas, de modo que una persona inexperta pueda ser capaz de interactuar naturalmente con ella, casi de manera inmediata o al menos requerir un mínimo de tiempo para



Figura 3.6. Valores ideales de aceleración para diferentes orientaciones del dispositivo

capacitación del manejo del dispositivo. El segundo punto a considerar es que se satisfagan ciertos criterios de seguridad, como los incluidos en las terminales de enseñanza de los robots. En el caso de la interfaz se incluye un botón con el cual se detiene el envío de información de un dispositivo a otro.

Con estos objetivos, se ha desarrollado una interfaz intuitiva que permite que, con solo presionar una flecha, el robot se desplace sobre la dirección indicada dentro del espacio de trabajo asignado. Se diseñaron 3 interfaces de trabajo como se ilustra en la Figura 3.7. En esta figura se puede ver cómo ha ido evolucionando la interfaz de usuario. La primera, de izquierda a derecha, solamente maneja un plano (y, z) y se utilizó para probar el canal de comunicación ya que, para cada punto deseado del efector final, se tenía que desplazar la esfera virtual a la coordenada deseada y había que presionar un botón para enviarla, es decir, el usuario realizaba manualmente el muestreo de puntos.

En la segunda interfaz ya se trabaja sobre un espacio tridimensional, pues se utilizaron las coordenadas (x, y, z). En este caso se le agregó un deslizador (slider) dentro de la interfaz, que permite el desplazamiento sobre el eje z. El envío de datos en esta segunda interfaz se hace de forma automática por medio de

un temporizador interno del Ipod, pero un problema que se percibió fue que la pantalla del iPod es muy pequeña para representar los casi 4 metros con que se puede desplazar el manipulador sobre su eje y. Por estos motivos se dejó a un lado la idea de limitar el espacio de trabajo del robot a la pantalla del dispositivo por medio de una esfera virtual.

Finalmente, la última interfaz se diseñó de tal manera que fuera lo mas intuitiva posible, respetando el amplio espacio de trabajo del manipulador. El modo de funcionamiento consiste en que, con solo presionar el botón de la flecha que indica la dirección deseada, el robot se mueve automáticamente hacia esa dirección con un incremento especificado previamente. Para detener el movimiento, solo se presiona el botón rojo.

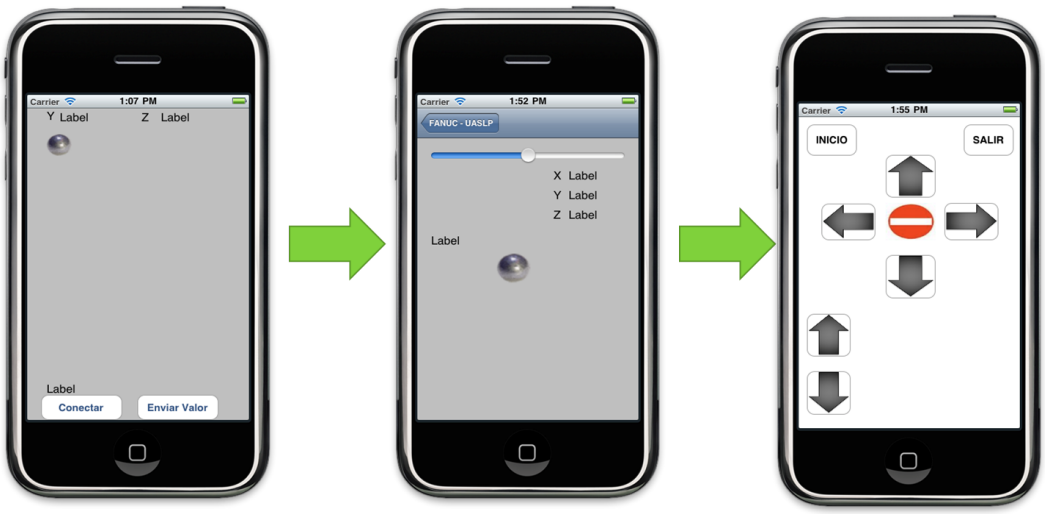


Figura 3.7. Evolución de interfaces para el dispositivo maestro

Para hacer más completa la interfaz de control, se propone el uso de los acelerómetros para el control de la orientación del efector final. Con este objetivo se escogieron las variables p y r, por razones que se explicarán el el siguiente capítulo.

El control de orientación, por medio de la interfaz, se realiza de manera sencilla. Se activa con presionar con dos dedos de manera simultanea cualquier parte de la pantalla del Ipod; inmediatamente en la pantalla se despliega una leyenda que dice *mandando orientación*, de esta forma se sabe que la interfaz se encuentra en modo orientación. Comenzado este proceso, se deberá mantener la inclinación del dispositivo hacia la dirección de la variable deseada, como se ilustra en la Figura 3.9. La idea que se manejó para hacer la interfaz más intuitiva, cuando se consigue la inclinación suficiente, en este caso una con valor mayor a 0.6 g en el eje de inclinación, la pantalla muestra una etiqueta donde indica que la variable de control se está modificando. En términos de seguridad se debe mantener la posición de inclinación deseada para que el efector final continúe con su movimiento de orientación alrededor del eje seleccionado. Para detener el movimiento de la orientación existen dos opciones: presionando el botón de paro que se encuentra en la parte inferior derecha de la pantalla o simplemente dejando el iPod en posición de inmóvil, como se muestra. Para salir

de modo orientación, simplemente se tiene que presionar cualquier otra flecha que indique un movimiento en la posición del robot.

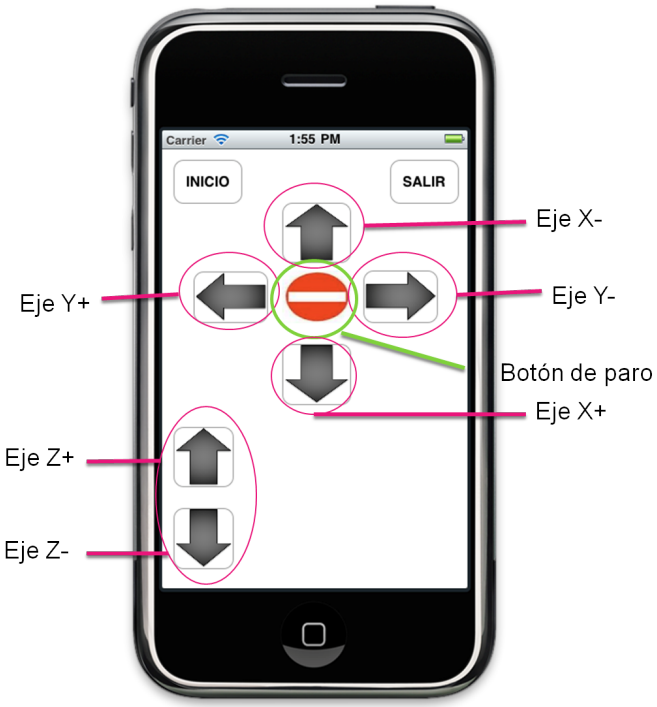


Figura 3.8. Interfaz diseñada para el sistema teleoperado



Figura 3.9. Dirección de la inclinación del dispositivo móvil para el control de la orientación

Capítulo 4

El problema de asimetría cinemática

En este capítulo se aborda el problema de la asimetría cinemática que tiene el sistema de teleoperación y la forma que se propuso para solucionar este problema de manera que el dispositivo teleoperado optimice su posición durante su trayectoria.

En el sistema teleoperado implementado, la asimetría de los dispositivos se puede ver desde varios puntos de vista. Entre ellos se destacan la dimensión de los dispositivos, el espacio de trabajo, los componentes, los grados de libertad, entre otros.

Cuando se habla de asimetría se refiere a que, tanto la configuración cinemática como el número de variables a controlar, es mayor que el número de variables disponibles. En este trabajo de tesis, por el lado del robot manipulador, se tienen 6 grados de libertad para ser controlados. Del dispositivo maestro, en principio, se cuenta solamente con las dos variables del plano de la pantalla. Para corregir este problema se propone un control cinemático.

4.1. Control Cinemático

Uno de los objetivos del control cinemático es establecer cuáles son las trayectorias que debe seguir cada articulación del robot a lo largo del tiempo para conseguir el objetivo fijado por el usuario, es decir, la posición del efector final. Del robot se tienen 6 grados de libertad, entonces es necesario el uso de 6 variables para el control, para de esta manera, corregir la asimetría. Se utilizaron x, y, z, w, p, r, como se muestra en la Figura 4.1, donde las tres primeras representan el desplazamiento de la punta del efector final a lo largo de cada eje coordenado, y las tres últimas el giro alrededor del respectivo eje.

Como se vio en el Capítulo 3, se ha diseñado una interfaz ergonómica e intuitiva, la cual permite controlar

el robot manipulador por medio de 5 variables de control, es decir, con la parte de posición de la interfaz, el manipulador se puede desplazar en un sistema coordenado (x, y, z). Al cambiar a modo orientación, únicamente se controlan las variables p y r, es decir el giro alrededor de los ejes y y z, respectivamente.

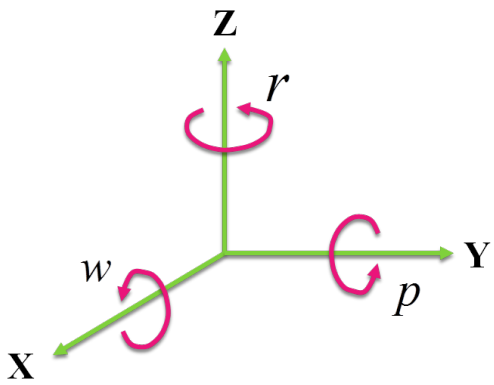


Figura 4.1. Sistema Coordenado fijo en la base del robot

Para una gran variedad de tareas industriales, como taladrado, pintura, desbastado, etc., se tiene que con cinco variables de control son suficientes. Debido a que la última variable de control es la orientación alrededor del eje perpendicular a la punta del efector final, la variación de este ángulo afecta a todas las variables de junta del manipulador, conservando la herramienta una posición final adecuada a las tareas industriales citadas, como se ilustra en la Figura 4.2.

En este caso, el robot es redundante y realiza satisfactoriamente la tarea. Con el objetivo de establecer el valor de las seis juntas, se busca que para la última variable, el algoritmo utilizado le asigne de forma automática su valor. Esta asignación se hace de tal modo que la configuración del robot sea óptima, como se explicará en la siguiente sección.

4.2. Obtención de la última variable de control

Retomando la ecuación (2.14), ésta se refiere a la matriz homogénea final que transforma las coordenadas de un punto desde el marco de referencia asociado a la herramienta al sistema coordenado de referencia colocado en la base del robot (x_0, y_0, z_0). Esta ecuación se puede reescribir de la siguiente manera:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} x_h \\ y_h \\ z_h \\ 1 \end{bmatrix} \tag{4.1}$$

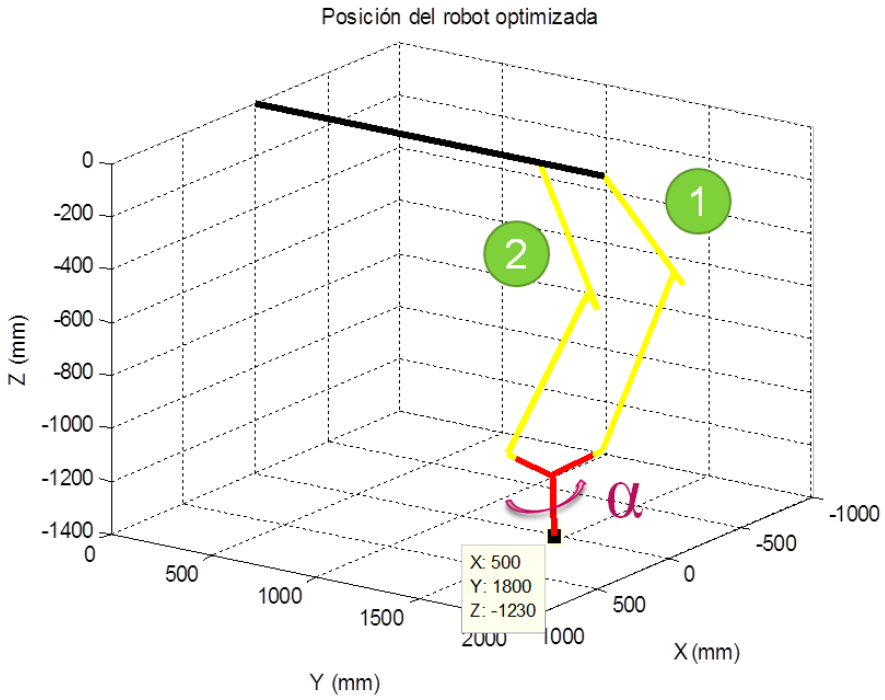


Figura 4.2. Orientación del robot, debida a la última variable de control

De la misma manera, la función que transforma las coordenadas de un punto desde el marco de la herramienta futuro (x'_h, y'_h, z'_h), al marco de la herramienta actual (x_h, y_h, z_h), se describe de la siguiente manera:

$$\begin{bmatrix} x_h \\ y_h \\ z_h \\ 1 \end{bmatrix} = \mathbf{R}_{(x,\alpha)} \begin{bmatrix} x'_h \\ y'_h \\ z'_h \\ 1 \end{bmatrix} \tag{4.2}$$

Ahora, se puede ver que en la figura 2.1 de nuestro manipulador, el eje coordenado que se encuentra perpendicular a la punta de la antorcha de nuestro robot es el eje x_h, por lo tanto, es necesaria una rotación alrededor del eje x, como lo describe la ecuación (4.3),

$$\mathbf{R}_{(x,\alpha)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) & 0 \\ 0 & \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}$$

De forma que, si el valor de este ángulo es $\alpha = 0$, se obtiene,

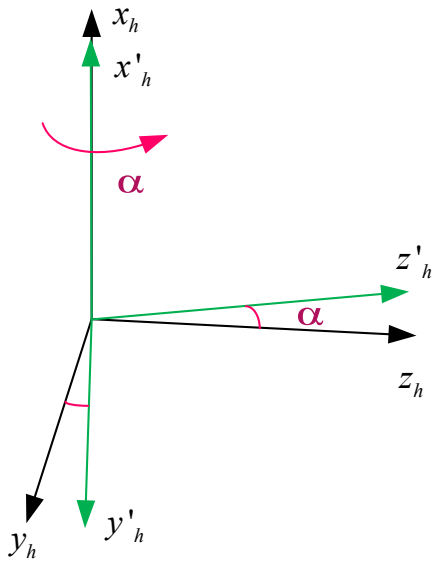


Figura 4.3. Rotación de un ángulo α alrededor del eje coordenado x

$$\mathbf{R}_{(x,0)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

que es la matriz identidad, es decir, el robot no tiene ningún cambio en su posición y orientación actual.

Finalmente, la relación que lleva desde la base del robot, hasta la nueva orientación del efector final, está definida por una post-multiplicación de la posición actual, como se muestra en la siguiente ecuación:

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \mathbf{TR}_{(x,\alpha)} \begin{bmatrix} x'_h \\ y'_h \\ z'_h \\ 1 \end{bmatrix} \tag{4.5}$$

4.3. Propuesta de la Función Objetivo

Ya que se tiene una variable de ajuste α, mediante la cual se afectan las 6 variables a controlar del robot, es necesario proponer una función objetivo unidimensional que dependa de esta variable. Partiendo de la condición de que la cinemática inversa permite conocer el valor de las 6 variables de junta, dependiendo de las variables cartesianas, se propone obtener la cinemática inversa a partir de la ecuación (4.5). De esta

manera se obtienen los nuevos valores de las 6 juntas, las cuales llevarán a la nueva posición y orientación final del manipulador, considerando la rotación alrededor del eje x_h.

La función a optimizar, es entonces, el valor del índice de manipulabilidad para una posición del robot afectada por el barrido de la variable de ajuste α dentro de un intervalo acotado.

Al optimizar la posición de robot, mediante el ajuste de la variable α, se está condicionando al manipulador a llegar de una sola forma a la posición deseada, debido a que se le asigna exactamente el valor de cada junta para esta posición. Al hacer esto, se está trabajando con un manipulador para una tarea no redundante, por esta razón, el índice de manipulabilidad que se utiliza es el de ecuación (2.47).

Este índice de manipulabilidad depende del valor de cada variable de junta del manipulador. De esta manera, se puede relacionar una función unidimensional que dependa de la variable α, y que a la vez modifique todos los valores de junta, para que el robot se encuentre en una posición óptima al momento de seguir una trayectoria. En la figura 4.4, se muestra el valor del índice de manipulabilidad para una posición cartesiana

Tabla 4.1. Posición inicial del manipulador

Variable	Valor
x	$500\ mm$
y	$1800\ mm$
z	$-1230\ mm$
w	$-50°$
p	$-85°$
r	$-20°$

inicial, indicada en la Tabla (4.1). Con un barrido del ángulo α que va de $-180°$ a $180°$, es decir, antes de obtener la cinemática inversa de la posición, ésta se multiplica por la matriz de rotación explicada anteriormente, donde α toma un valor diferente en cada evaluación. Una vez realizada esta multiplicación, se procede a obtener la cinemática inversa del manipulador para obtener los valores de las variables de junta y así evaluar el índice de manipulabilidad.

Cómo se explicó en la sección anterior, cuando el valor del ángulo α es cero, el índice de manipulabilidad y el valor de las juntas no se ven afectadas. Conforme el valor del ángulo α va incrementando o disminuyendo, el índice de manipulabilidad y el valor de las juntas cambia. Se puede observar que, para el caso en cuestión, el valor del ángulo α en el cual el índice de manipulabilidad tiene un máximo local, que se encuentra en $-54°$.

Se puede ver que es posible obtener un valor máximo del índice de manipulabilidad, a través de una variación del ángulo α. Entonces, se propone que la función unidimensional objetivo sea el índice de manipulabilidad ya que permite minimizar el riesgo de caer en una singularidad.

Finalmente, al querer maximizar la función del índice de manipulabilidad, se tiene un problema de

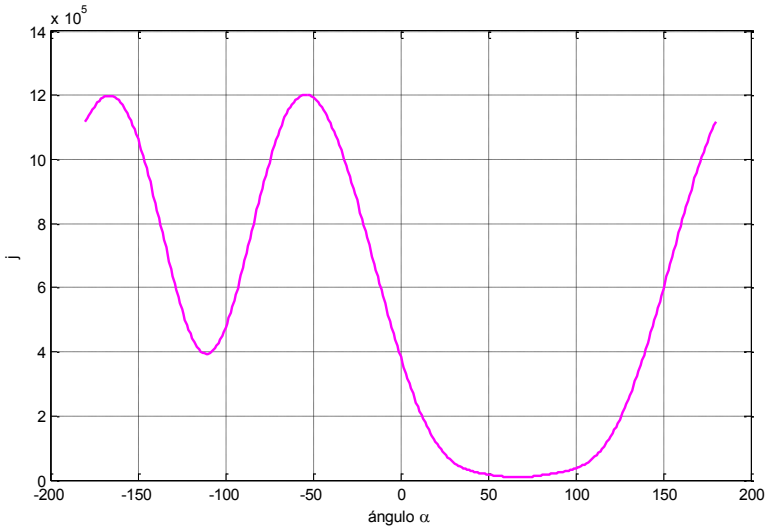


Figura 4.4. Variación del índice de manipulabilidad por la variable α

optimización para seleccionar la variable α. Este problema se aborda en la siguiente sección.

4.4. Optimización de la Función Objetivo

La optimización tiene que ver con la determinación del mejor resultado, o solución óptima, de un problema. Así, en el contexto del modelado, se les llama con frecuencia modelos prescriptivos, puesto que sirven para señalar un curso de acción o el mejor diseño.

Los ingenieros constantemente tienen que diseñar dispositivos y productos que realicen tareas de manera eficiente. Al hacerlo de esta manera, está restringido por las limitaciones del mundo físico, además de mantener los costos bajos. Así, los ingenieros se enfrentan a problemas de optimización que equilibren el funcionamiento y las limitaciones.

Una forma de clasificar los problemas de optimización, es según su dimensionalidad. En general se dividen en unidimensionales y multidimensionales. Como su nombre lo indica, los primeros involucran funciones que dependen de una sola variable independiente, es decir, la búsqueda consiste dentro del barrido de la variable en un rango acotado. Los problemas multidimensionales implican funciones que dependen de dos o más variables independientes. Otra forma en la que se pueden dividir los problemas de optimización es en métodos cerrados y abiertos. La búsqueda de la sección dorada es un método cerrado [25], que depende de los valores iniciales que encierran un valor óptimo.

4.4.1. Búsqueda de la sección dorada

Los principales factores a considerar cuando se programa un método numérico con una función compleja, es hacerlo rápido, de forma eficiente, y que ocupe poca memoria. Todo en general se resume al costo de evaluación de la función, es decir, evaluar la función lo menos posible.

Se selecciono el método de búsqueda de la sección dorada [24] por varias razones, entre ellas están:

- la búsqueda de un algoritmo sencillo debido a que, tanto en el iPod como en el controlador del robot, no se cuenta con una gran capacidad de procesamiento.
- es una técnica de búsqueda para una sola variable, en este caso la ya definida α.
- es una técnica sencilla, ya que no necesita derivadas como el método de Newton Rapson, únicamente la evaluación de la función. Esto se traduce a una menor demanda al procesador pues el índice de manipulabilidad a maximizar tiene un gran número de términos.
- para este método numérico en cada iteración solo se calcula un nuevo valor, los otros valores se reasignan; esto tiene como consecuencia el ahorro tiempo y esfuerzo computacional.

Se definen x_i y x_f, como los límites inferior y superior respectivamente, del intervalo. Sin embargo, a diferencia de la bisección, se necesita una nueva estrategia para encontrar un máximo dentro del intervalo. En vez de usar solamente dos valores de la función, los cuales son suficientes para detectar un cambio de signo y, por lo tanto, un cero. En este caso, se necesitarán tres valores de la función para detectar si hay un máximo. Así, hay que escoger un punto más dentro del intervalo. Después hay que tomar un cuarto punto. La prueba para el máximo podrá aplicarse para determinar si éste se encuentra dentro de los primeros tres o de los últimos tres puntos.

La clave para hacer eficiente este procedimiento es la adecuada elección de los puntos intermedios. Como en la bisección, la meta es minimizar las evaluaciones de la función reemplazando los valores anteriores con los nuevos. Esta meta se puede alcanzar especificando que las siguientes dos condiciones se satisfagan (Ver Figura 4.5):

$$\ell_0 = \ell_1 + \ell_2 \tag{4.6}$$

$$\frac{\ell_1}{\ell_0} = \frac{\ell_2}{\ell_1} \tag{4.7}$$

La primera condición especifica que la suma de las dos sublongitudes ℓ_1 y ℓ_2 debe ser igual a la longitud original del intervalo. La segunda indica que el cociente o razón entre las longitudes debe ser igual. La ecuación (4.6) se sustituye en la (4.7).

$$\frac{\ell_1}{\ell_1 + \ell_2} = \frac{\ell_2}{\ell_1} \tag{4.8}$$

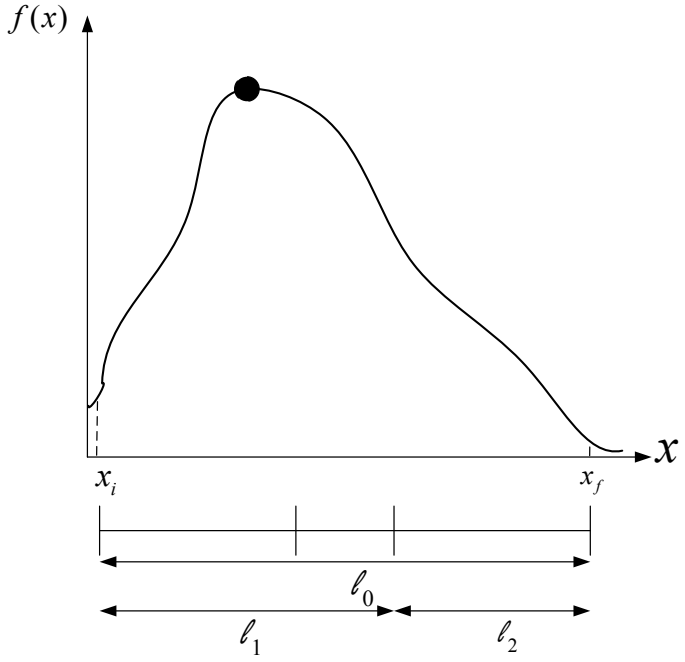


Figura 4.5. Condiciones necesarias para la sección dorada

Si se toma el recíproco y $R = \frac{\ell_2}{\ell_1}$, se llega a

$$1 + R = \frac{1}{R} \tag{4.9}$$

o

$$R^2 + R - 1 = 0 \tag{4.10}$$

de la cuál se obtiene la raíz positiva

$$R = \frac{-1 + \sqrt{1 - 4(-1)}}{2} = \frac{\sqrt{5} - 1}{2} = 0.61803.... \tag{4.11}$$

Este valor, que se conoce desde la antigüedad, se llama *razón dorada*. Como permite encontrar el valor óptimo en forma eficiente, es el elemento clave del método de la sección dorada.

El metodo comienza con dos valores iniciales, x_i y x_f, que contienen un extremo local de la función objetivo $f(x)$. Después, se eligen dos puntos interiores x_1 y x_2, de acuerdo a la razón dorada,

$$\begin{aligned} d &= R(x_f - x_i) \\ x_1 &= x_i + d \\ x_2 &= x_f - d \end{aligned}$$

La función objetivo se evalúa en estos dos puntos interiores. Dos casos pueden presentarse:

1. Si, como en el caso de la Figura 4.6, $f(x_1) > f(x_2)$, entonces el dominio de x a la izquierda de x_2, de x_i a x_2, se puede eliminar, ya que no contiene el máximo. En este caso, x_2 será el nuevo x_i en la siguiente vuelta.

2. Si $f(x_2) > f(x_1)$, entonces el dominio de x a la derecha de x_1, de x_1 a x_f podrá eliminarse. En este caso, x_1 será el nuevo x_f en la siguiente iteración.

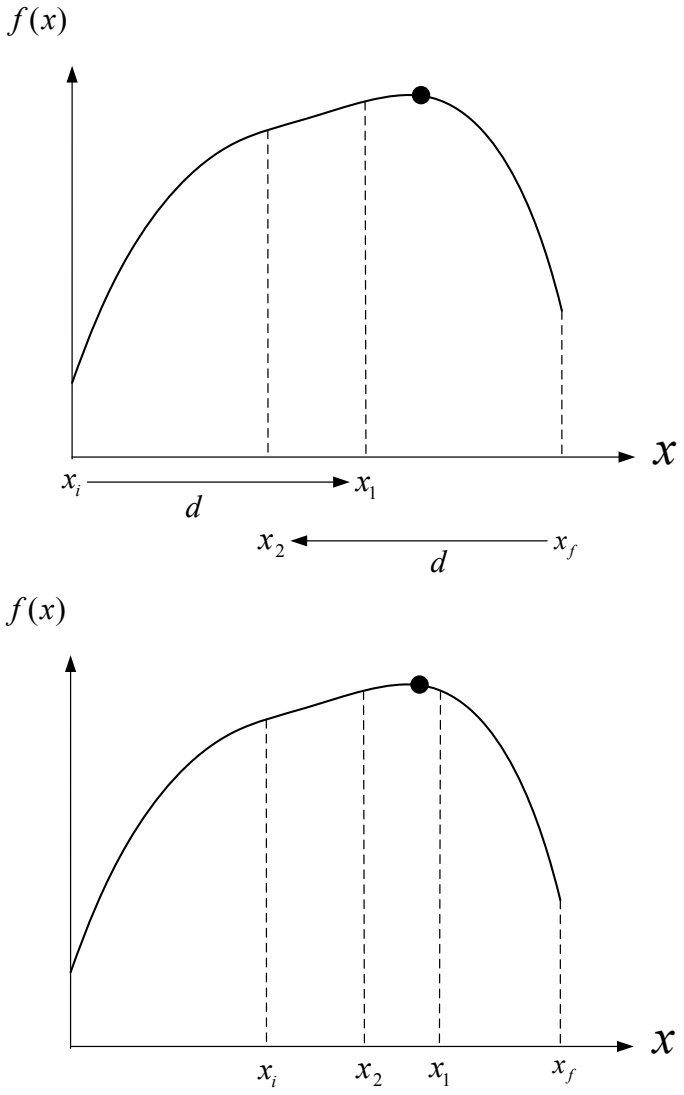


Figura 4.6. Comienzo del algoritmo de búsqueda de la sección dorada

Ahora, esta es la ventaja real del uso de la razón dorada. Debido a que los x_1 y x_2 originales se han escogido mediante la razón dorada, no se tienen que recalcular todos los valores de la función objetivo en la siguiente

iteración. Por ejemplo, en el caso de la Figura (4.6), el anterior x_1 será el nuevo x_2. Esto significa que ya se tiene el valor para el nuevo $f(x_2)$, puesto que es el mismo valor de la función en el anterior x_1.

Para completar el algoritmo, ahora sólo se necesita determinar el nuevo x_1. Esto se realiza usando la misma proporcionalidad que antes,

$$x_1 = x_i + \frac{\sqrt{5}-1}{2}(x_f - x_i)$$

Un procedimiento similar podría usarse en el caso de que el óptimo caiga del lado izquierdo del subintervalo.

Conforme las iteraciones se repiten, el intervalo que contiene el extremo se reduce rápidamente. De hecho, en cada iteración el intervalo se reduce en un factor de la razón dorada (aproximadamente 61.8 $\%$). Esto significa que después de 10 iteraciones, el intervalo se acorta aproximadamente en $0.8\,\%$ de su longitud inicial. Después de 20 iteraciones, se encuentran en $0.0066\,\%$.

Para condicionar la salida del algoritmo de optimización se propone la función de error

$$e = (1 - R)\left|\frac{x_f - x_i}{x_{opt}}\right| 100\,\% \tag{4.12}$$

Se ha definido en el algoritmo un error suficiente de 0.001, por la razón de que en el controlador de robot solo se tiene capacidad de manejar 3 decimales. Es decir, un error más pequeño no afectaría la posición objetivo del manipulador y solo incrementaría el número de iteraciones en nuestro algoritmo.

En la Figura 4.7 se presenta un diagrama de flujo, para la búsqueda de la sección dorada en la máximización.

La razón por la que se ha hecho énfasis en reducir las evaluaciones de la función para la búsqueda de la sección dorada, se puede resumir a dos casos:

1. *Muchas evaluaciones* En este caso el algoritmo de búsqueda de la sección dorada puede ser parte de otros cálculos. Entonces, es llamado muchas veces, por lo tanto, mantener el número de evaluaciones de la función en un mínimo ofrecería dar grandes ventajas en este caso.

2. *Evaluaciones que toman mucho tiempo* Se debe tener en cuenta que una función puede llegar a ser muy compleja, como la del índice de manipulabilidad y, además, consumir mucho tiempo de evaluación. Entonces un método que minimice tales evaluaciones resultará provechoso.

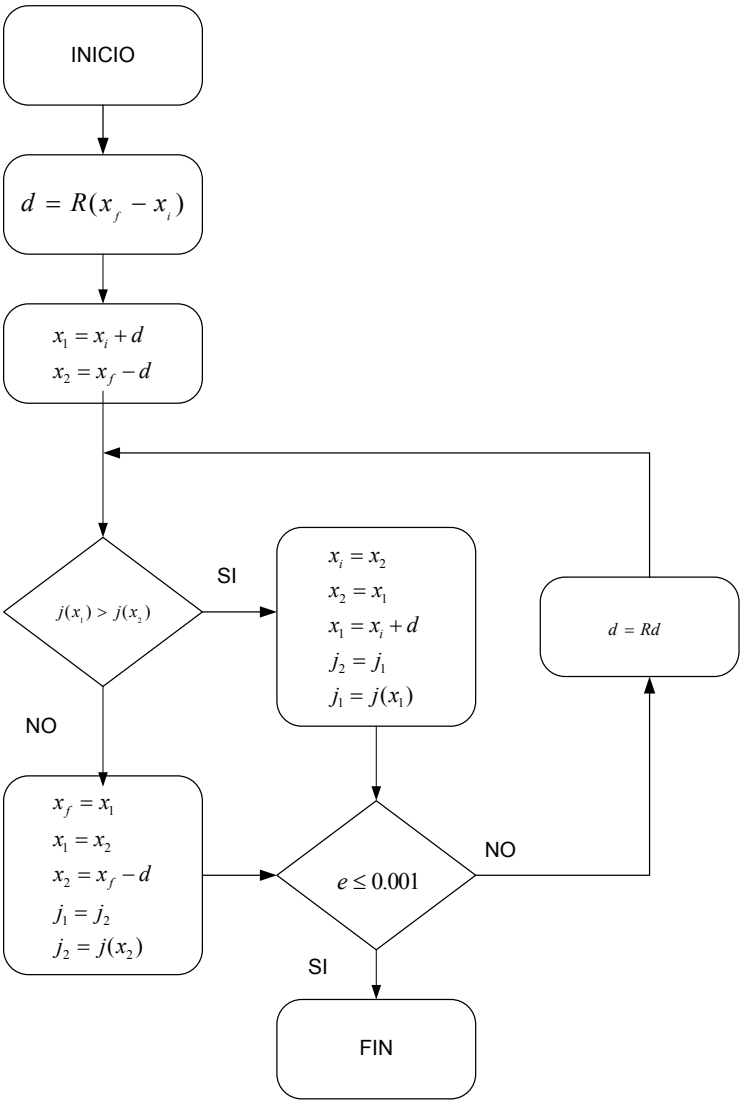


Figura 4.7. Algoritmo para la búsqueda de la sección dorada

Capítulo 5

Resultados experimentales

5.1. Plataforma experimental

La implementación del sistema de teleoperación unilateral se realizó en el laboratorio de robótica de la Facultad de Ingeniería, de la Universidad Autónoma de San Luis Potosí. La plataforma experimental se puede separar en tres elementos importantes los cuales consisten en: el diseño de una interfaz dentro de un dispositivo móvil, un canal de comunicación y el algoritmo de control que se ejecuta dentro del controlador del robot industrial.

Para el desarrollo de la interfaz, se contó con una computadora MacBook, un Ipod Touch de 32 Gb, así como con la licencia de desarrollador de Apple, y con el SDK (Software Development Kit), que permite cargar las aplicaciones realizadas en el dispositivo. El SDK cuenta con dos programas principales para el desarrollo de interfaces, se utilizó el Xcode Versión 3.2.4 en 64-bits, para la parte del modelo. Mientras que el programa Interface Builder, es el encargado del diseño visual de la aplicación.

Los datos de posición y orientación serán enviados del Ipod al robot manipulador, por tanto los datos enviados deben ser transmitidos de una manera confiable, pero también tomando en cuenta que el retraso de la tasa de transmisión no sea demasiado lenta. Un socket es un punto de comunicación por el cual un proceso puede emitir o recibir información, éste queda definido por una dirección IP, un protocolo de transporte y un número de puerto.

Gracias al protocolo TCP/IP, los sockets tienes las siguientes propiedades:

- La fiabilidad de la transmisión. Ningún dato transmitido se pierde.
- La conservación del orden de los datos. Los datos llegan en el orden en el que han sido emitidos.

- La no duplicación de datos. Sólo llega a destino un ejemplar de cada dato emitido.

- La comunicación en modo conectado. Se establece una conexión entre dos puntos antes del principio de la comunicación (es decir, se establece un circuito virtual). A partir de entonces, una emisión desde un extremo está implícitamente destinada al otro extremo conectado.

Todas estas propiedades son indispensables para el envío correcto de información de posición hacia el robot industrial. Los sockets permiten implementar una arquitectura cliente-servidor. Una vez establecida la conexión entre un servidor y un cliente a través de dos sockets, los dos procesos pueden intercambiar flujos de información.

Por estas razones, para el canal de comunicación entre el dispositivo móvil (maestro) y el controlador del robot (esclavo) se hizo uso de la programación de sockets para la creación de un sistema cliente-servidor. En primer lugar se crea el programa servidor en el controlador del robot, haciendo uso del puerto 59002. Después se desarrolla el programa cliente en el dispositivo maestro, indicando la dirección IP del robot y el número de puerto a donde debe enviar la información.

Respecto a la programación del robot industrial, para recibir las posiciones y realizar el movimiento hacia éstas, se creó un algoritmo de control por medio del lenguaje KAREL, utilizando OlpcPRO que es un software de última generación para el desarrollo de programación fuera de línea, para los robots FANUC [26].

En la Figura 5.1 se puede observar, a grandes rasgos, los componentes del sistema teleoperado implementado. Lo que se encuentra dentro de las lineas punteadas es lo que se ejecuta por cada muestra o cadena de datos enviados del Ipod hacia el manipulador.

El sistema teleoperado comienza cuando el operador presiona la tecla Inicio. En ese momento, el robot obtiene una posición inicial x, y, z, w, p, r. Cuando el operador activa cualquiera de las teclas que posicionan u orientan el robot, la cadena de datos de posición y orientación cambian a $\bar{x}, \bar{y}, \bar{z}, \bar{w}, \bar{p}, \bar{r}$, en ese instante, por medio de la cinemática inversa, se obtienen los valores de juntas $\theta_1, ..., \theta_6$, los cuales son utilizados para optimizar la posición del robot, como se explicará mas adelante. Una vez optimizada la posición del manipulador se obtienen los nuevos valores de juntas $\theta_{1o}, ..., \theta_{6o}$, los cuales son enviados por medio de sockets hacia el algoritmo del controlador del robot. Este algoritmo recibe y descompone la cadena de datos, desplazando al manipulador a la posición óptima deseada.

Una vez que el operador activa cualquiera de las teclas que posicionan u orientan el robot, un temporizador interno del Ipod se activa. Este temporizador envía, cada 50 ms, una nueva cadena de datos la cual aumenta el valor de la variable seleccionada con cada iteración. Este proceso se repite hasta que el usuario presiona el botón de paro ó finaliza la aplicación.

Finalmente, como medida de precaución antes de probar una trayectoria directamente con el robot industrial

se diseñó, con el modelo cinemático directo e inverso del manipulador, un simulador en el programa MatLab Vesion 7.9, para emular el seguimiento de trayectorias del robot.

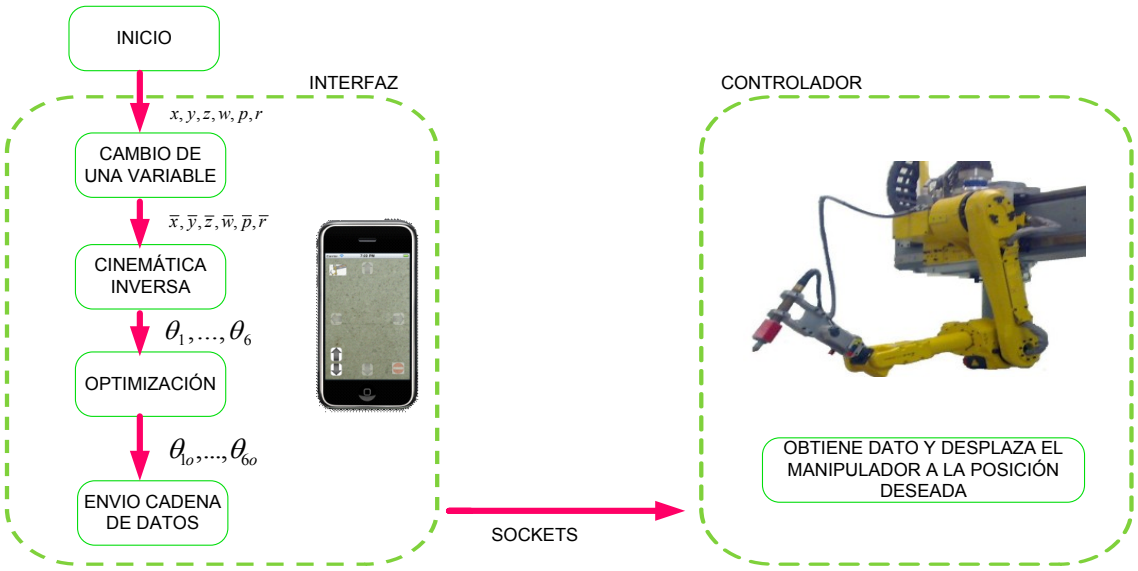


Figura 5.1. Sistema teleoperado implementado

5.2. Trayectoria Preplaneada

Para verificar el correcto posicionamiento del efector final del manipulador, se realizaron diferentes pruebas de trayectorias para verificar el posicionamiento, fidelidad, y tiempo de muestreo del sistema teleoperado.

En una prueba inicial se utilizó una trayectoria solo de posición, que abarca las tres coordenadas cartesianas. Los resultados obtenidos se presentan en la Figura 5.2, donde se aprecia cómo el manipulador esclavo realiza un correcto seguimiento de la trayectoria indicada por el maestro.

Para la realización de esta trayectoria, el manipulador Maestro realiza el envío de la cadena de datos cada 50 milisegundos, donde en cada muestreo, el manipulador se desplaza 5 mm en la dirección seleccionada. Por parte del controlador del robot, se tiene una variable que mide el tiempo desde que se realiza la captura del dato, se procesa y el manipulador se desplaza a la posición deseada. La Figura 5.3 muestra los tiempos de muestreo de la trayectoria anterior.

Los valores estadísticos de los tiempos de muestreo de la trayectoria de la Figura 5.2, se muestran en la Tabla 5.1. Se observa que el promedio de los valores de los tiempos de muestreo es alrededor de los 204 ms, es decir, el procesamiento, envío de datos y movimiento del robot, tarda cerca de 150 ms en realizarse.

Otra prueba que se realizó, fue el seguimiento de una trayectoria simple del robot, que se establece sobre un

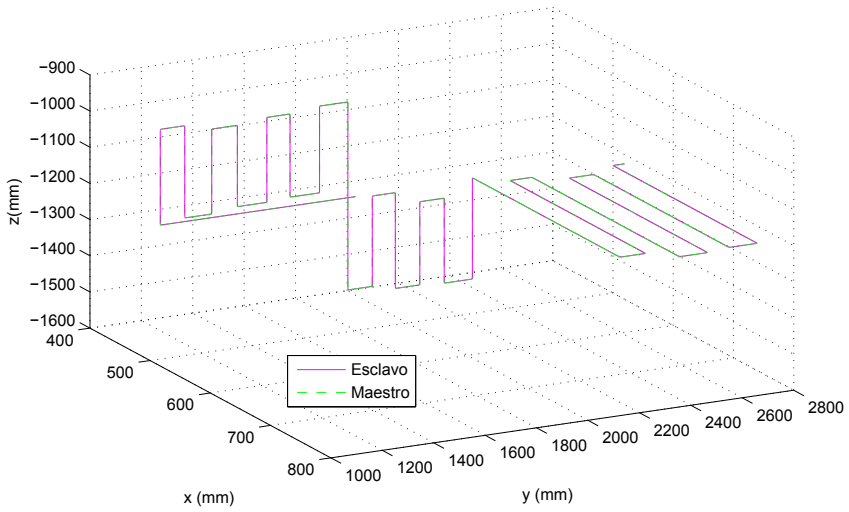


Figura 5.2. Trayectoria del manipulador FANUC M16-iB 20T

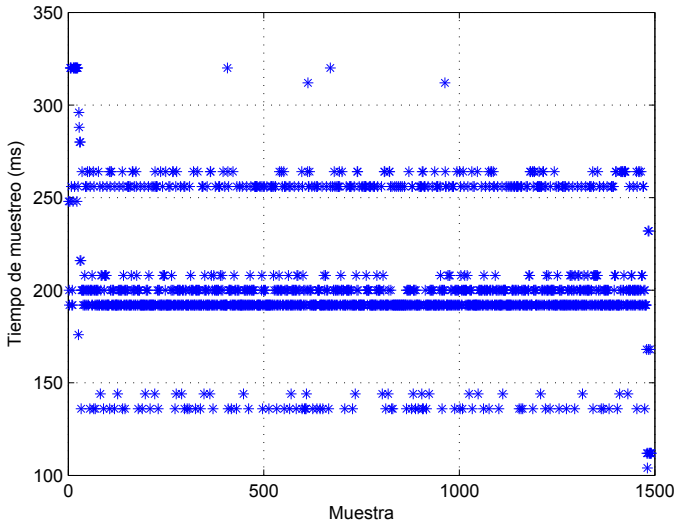


Figura 5.3. Tiempo de muestreo de una trayectoria

Tabla 5.1. Valores estadísticos de la comunicación del sistema teleoperado

Medida	Valor
Media	204.8719 ms
Varianza	1144.1 ms
Desviación estándar	33.8252 ms

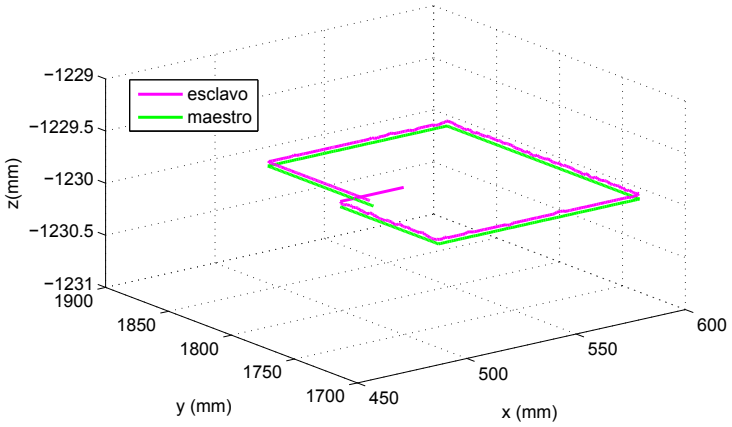


Figura 5.4. Trayectoria perpendicular al eje z

plano, en este caso, sobre el plano (x, y), dejando el eje z fijo con un valor de -1230 mm. Los resultados se muestran en la Figura 5.4, en la cual, se observa el correcto seguimiento del esclavo, respecto al manipulador maestro.

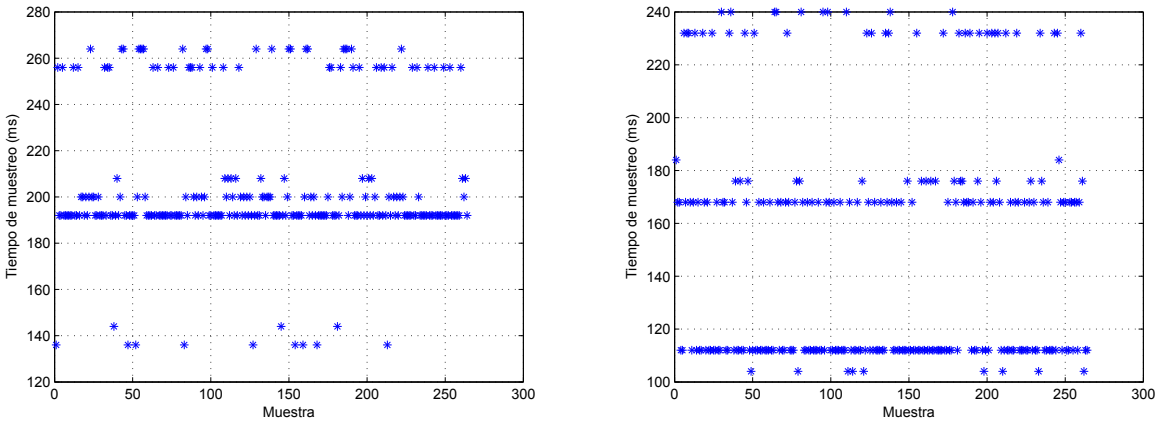
Sin embargo, se puede apreciar cómo la ubicación del efector final del robot manipulador sobre el eje fijo z, varía continuamente durante la trayectoria deseada. Estos resultados son mostrados en la Tabla 5.2, la cual nos indica que la variación de posición del manipulador esclavo, con respecto al maestro es alrededor de 0.005 mm. Esto nos dice que el sistema teleoperado implementado tiene un seguimiento con un error de posicionamiento casi nulo.

Tabla 5.2. Valores estadísticos de la posición del eje z, sobre una trayectoria simple

Medida	Valor
Media	$-1229.96228\ mm$
Varianza	$0.00003\ mm$
Desviación estándar	$0.00578\ mm$

Una prueba interesante realizada para la trayectoria de la Figura 5.4, fue cambiar el valor que aumenta la variable seleccionada en cada iteración. La gráfica del lado izquierdo de la Figura 5.5, muestra el tiempo de muestreo que realiza la trayectoria con un aumento de 5 mm sobre la variable seleccionada en cada iteración. En la gráfica del lado derecho se observa el tiempo de muestreo de la trayectoria con un aumento de 1 mm.

Los resultados de los tiempos de muestreo se muestran en la Tabla 5.3, donde se observa que en el desplazamiento de 1 mm en cada iteración del robot, se muestra el retardo en la comunicación más pequeño, alrededor de 50 ms. Con esto, se puede concluir que el tiempo que tarda el manipulador en desplazarse a la posición deseada afecta en el retardo de la comunicación, debido a que en la comunicación por sockets, el cliente no puede enviar un segundo dato, hasta que el servidor no confirme la correcta ejecución del dato



(a) Desplazamiento a 5 mm

(b) Desplazamiento a 1 mm

Figura 5.5. Tiempos de muestreo de una trayectoria con diferente taza de desplazamiento

anterior. Esto se logra al término del posicionamiento del robot.

Tabla 5.3. Valores estadísticos de la comunicación del sistema teleoperado

Medida	Desplazamiento (1mm)	Desplazamiento (5mm)
Media	151.124 ms	205.515 ms
Varianza	2126.3 ms	915.596 ms
Desviación estándar	46.112 ms	30.258 ms

Se debe considerar que, aunque el desplazamiento de 1mm por cada iteración hace el retardo de comunicación mas pequeño, visualmente el movimiento del manipulador es más lento, ya que avanza una distancia muy pequeña. En cambio, si el desplazamiento es de 5mm, la cantidad de datos se reduce por el retraso de la comunicación, pero el robot se desplaza mas rápido durante la trayectoria. Se puede concluir que en este sistema teleoperado se encuentran relacionadas la velocidad del robot, con el retardo en la comunicación.

5.3. Optimización de Posición

La optimización de la posición del robot durante una trayectoria, está basada en la maximización del valor del índice de manipulabilidad. El índice de manipulabilidad se obtiene calculando los valores de juntas mediante la cinemática inversa de la posición cartesiana a la que se desplaza el robot. En la Figura 5.6 se muestra el valor del índice de manipulabilidad, para la trayectoria de la Figura 5.2.

En la gráfica se observa el valor del índice de manipulabilidad para cada muestra; una muestra es una cadena de datos que envía la posición deseada al controlador del robot. La leyenda j real, se refiere al valor del índice

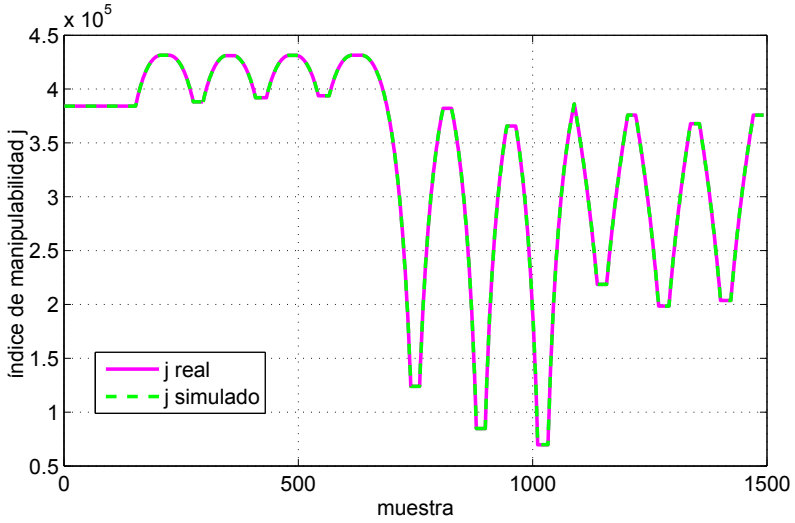


Figura 5.6. Índice de manipulabilidad de una trayectoria planeada

de manipulabilidad tomando los valores de las juntas directamente del robot, sin incluir la optimización de la variable α. La leyenda j simulado, se refiere a los valores de las juntas obtenidas mediante un algoritmo implementado en MATLAB, para validar la cinemática inversa del robot.

Para maximizar el valor del índice de manipulabilidad en cada muestra, se parte de la cadena de datos a donde se quiere colocar el manipulador (x, y, x, w, p, r); con estos datos se obtiene la matriz de rotación final que va desde la base del robot hasta el extremo del efector final. Esta matriz de rotación se posmultiplica por una rotación alrededor del eje x con un valor α óptimo que se adquiere por el método de búsqueda de la sección dorada. A esta posición final reorientada por el ángulo α se le calcula la cinemática inversa y se obtienen los nuevos valores de juntas $\theta_1, ..., \theta_6$ optimizados.

La Figura 5.7, muestra la diferencia entre los valores del índice de manipulabilidad optimizados y sin optimizar, para la trayectoria de la Figura 5.2.

Se probaron diversas trayectorias en el sistema teleoperado, dentro del espacio de trabajo del manipulador. Se encontró que existen algunas trayectorias donde la posición del robot se encuentra comprometida, por ejemplo, cerca de los límites inferiores en el eje z, como la que se muestra en la Figura 5.8. En donde se desplazó al manipulador en dirección negativa sobre el eje z, partiendo desde la posición inicial mostrada en la Tabla 4.1.

Si se analiza la gráfica del índice de manipulabilidad para la posición optimizada de la trayectoria definida sobre el eje z, se puede observar que existen ciertas discontinuidades como se observa en la Figura 5.9. Esto genera un cambio de postura abrupta en el robot, es decir, el robot mantiene la posición del punto en el extremo del efector final, pero su optimización alrededor de la punta del extremo de la herramienta cambia constantemente, lo que hace que el robot cambie su posición justamente en cada discontinuidad de

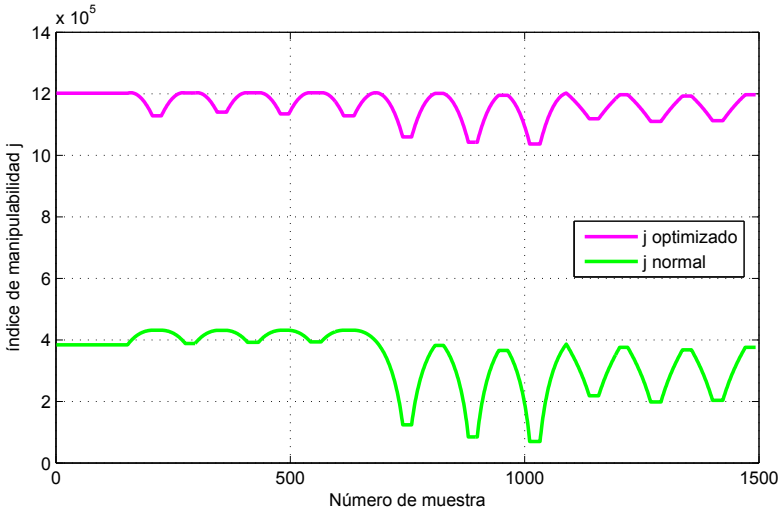


Figura 5.7. Comparación del índice de manipulabilidad para una trayectoria

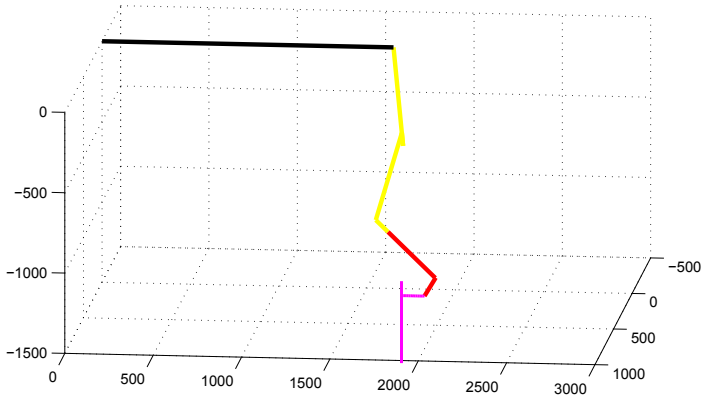


Figura 5.8. Trayectoria del manipulador sobre el eje z

la gráfica.

Se sabe que el parámetro que afecta directamente el valor del índice de manipulabilidad es el ángulo α. Si se gráfica el valor de este ángulo, en cada muestra o cadena de datos, se obtiene el resultado que se muestra en la Figura 5.10, donde se puede observar cómo es el cambio abrupto del valor de α.

Para verificar la razón por la que el ángulo optimizado α cambia abruptamente, se analizó una muestra de datos de cada cambio considerable del valor de este ángulo. Los resultados obtenidos se muestran en la Tabla 5.4.

Debido a la longitud de los eslabones del manipulador, le es imposible alcanzar algunas posiciones. En la

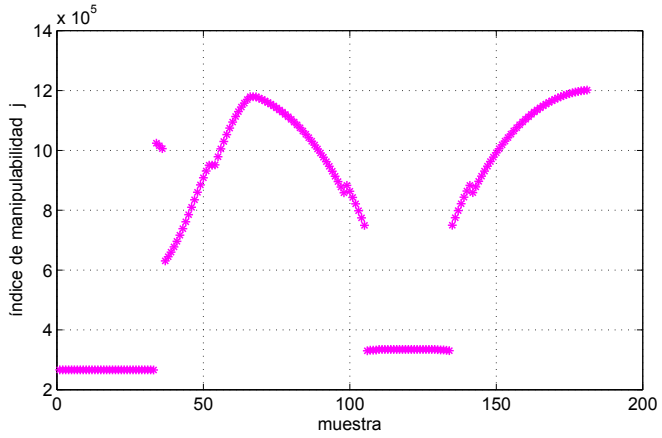


Figura 5.9. Índice de manipulabilidad optimizado

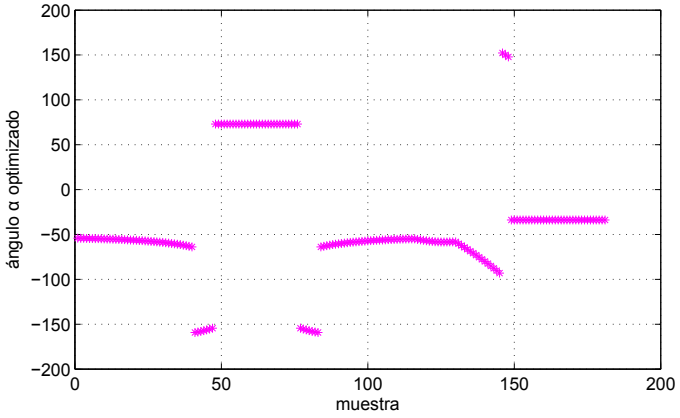
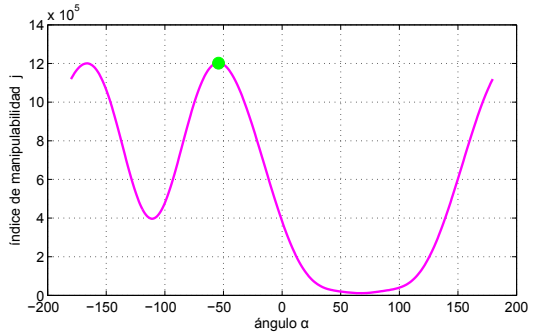


Figura 5.10. Valor del ángulo α durante una trayectoria

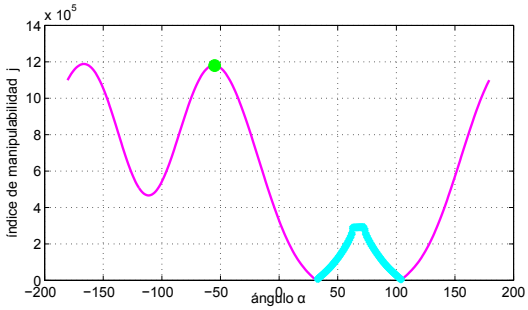
sección de la cinemática inversa en la ecuación 2.24, cuando el valor de D es mayor que 1, el ángulo β se convierte en un número complejo, lo que hace que el robot no alcance la posición deseada.

Uno de los primeros resultados obtenidos es relacionado a la selección de una posición inicial óptima del manipulador. Como se observa en la Figura 5.11(a), esta gráfica muestra la posición inicial del manipulador. Se eligió esta posición debido a que el manipulador puede alcanzar cualquier posición a partir de cualquier valor del ángulo α.

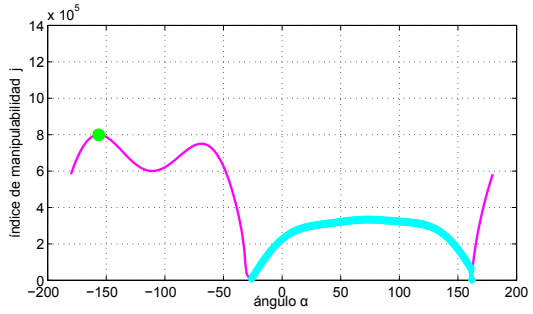
El punto verde que se encuentra en las gráficas de la Figura 5.11, es el valor óptimo del ángulo α que arroja como resultado el algoritmo de optimización de la búsqueda de la sección dorada; estos valores de ángulos se describen el la Tabla 5.4. Las trayectorias que están de color azul, son posiciones no alcanzados por el manipulador, es decir, números imaginarios que afectan el algoritmo de optimización del ángulo α. Se dice que afectan el algoritmo, debido a que solo se toma la parte real de estos números, de esta manera, se



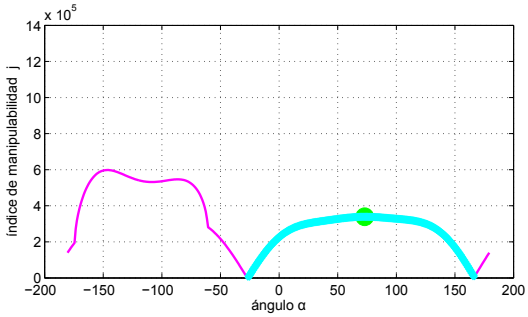
(a) Muestra 1, α=-54.196^{o}



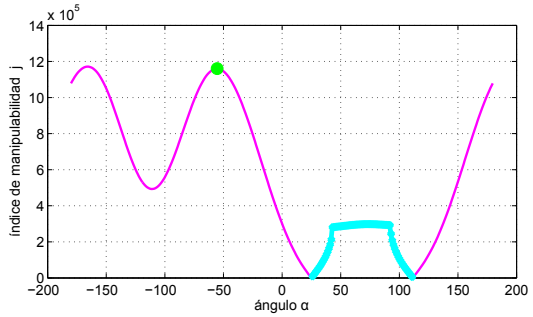
(b) Muestra 18, α=-55.992^{o}



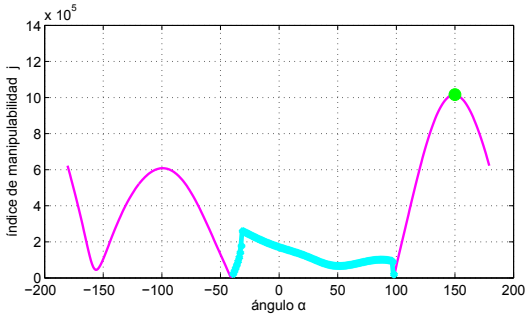
(c) Muestra 45, α=-156.385^{o}



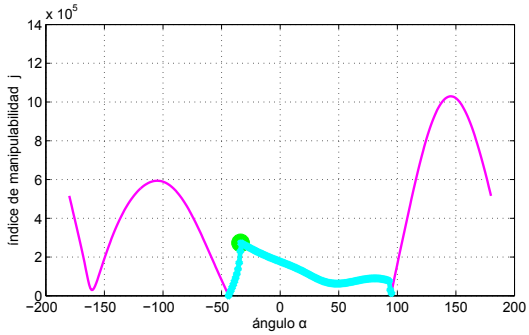
(d) Muestra 64, α=72.990^{o}



(e) Muestra 110, α=-55.381^{o}



(f) Muestra 147, α=150.017^{o}



(g) Muestra 165, α=-33.812^{o}

Figura 5.11. Cambios del índice de manipulabilidad en una trayectoria definida

puede decir que el algoritmo evalua datos incompletos que afectan el valor de salida, dando como resultado máximos locales no deseados. Lo que se propone para corregir este problema, es tratar de que el algoritmo no

Tabla 5.4. Tabla comparativa de valores optimizados

Figura	No. de muestra	Posición (x, y, z, w, p, r)	α	j
5.11(a)	1	$(500, 1800, -1230, -50, -85-, 20)$	-54.196^{o}	1201627
5.11(b)	18	$(500, 1800, -1400, -50, -85-, 20)$	-55.992^{o}	1135383
5.11(c)	45	$(500, 1800, -1670, -50, -85-, 20)$	-156.385^{o}	798657
5.11(d)	64	$(500, 1800, -1740, -50, -85-, 20)$	72.990^{o}	339137
5.11(e)	110	$(500, 1800, -1316, -50, -85-, 20)$	-55.381^{o}	1160372
5.11(f)	147	$(500, 1800, -1320, -50, -49-, 31)$	150.017^{o}	1016017
5.11(g)	165	$(500, 1800, -1320, -50, -49-, 37)$	-33.812^{o}	273133

evalúe las posiciones con números complejos para que el valor de α no cambie de forma brusca. Por lo tanto, se encontró una posición inicial óptima que permita encontrar el máximo valor del índice de manipulabilidad evaluando α desde un barrido de ángulos de -180^{o} a 180^{o}. Como esta primera posición ya es óptima para el robot, para las siguientes posiciones, el algoritmo de la búsqueda de la sección dorada tiene nuevos límites. En lugar de hacer el barrido de -180^{o} a 180^{o}, ahora se acota el rango de búsqueda el cual tiene como centro el valor α óptimo anterior y un ancho de $\pm 10^{o}$. De esta manera se evita que el algoritmo evalue posiciones no alcanzables por el robot.

Los resultados de la aplicación de este nuevo filtro son muy favorables. Como se muestra en la Figura 5.12, la trayectoria de color verde ahora es continua, es decir, los valores del ángulo optimizado α no cambian abruptamente; de esta manera el robot adquiere un movimiento más suave durante la trayectoria.

De la misma forma que el valor del ángulo optimizado es continua, lo mismo sucede con el valor del índice de manipulabilidad. En la gráfica 5.13 se muestra la comparación de los valores del índice de manipulabilidad con las diferentes ventanas de valores.

Se puede concluir que, con este nuevo algoritmo, el robot ejecuta su trayectoria óptima de manera suave. El sacrificio que se hace es no garantizar el valor máximo global del índice de manipulabilidad, sino que se enfoca en un máximo local a lo largo de la trayectoria.

5.4. Implementación de control cinemático de impedancia basado en posición

Para cumplir el objetivo del diseño de un sistema teleoperado capaz de ser implementado en varios robots fácilmente, se decidió integrar el sistema de teleoperación unilateral con optimización de la posición, con

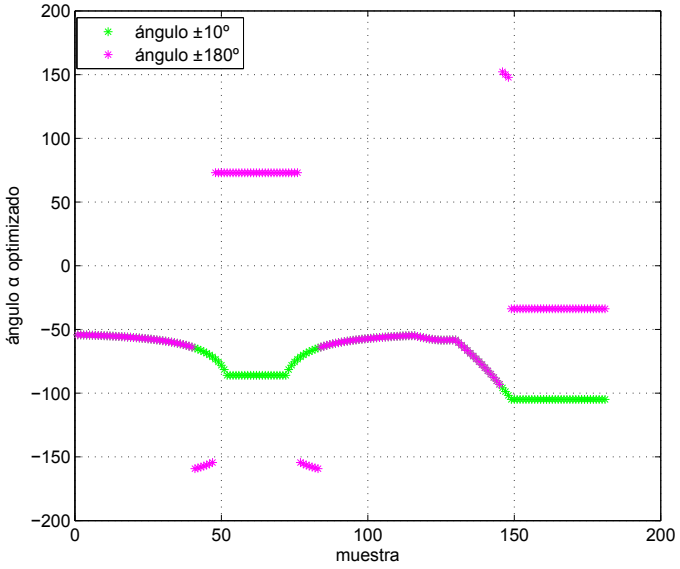


Figura 5.12. Corrección del valor del ángulo α optimizado

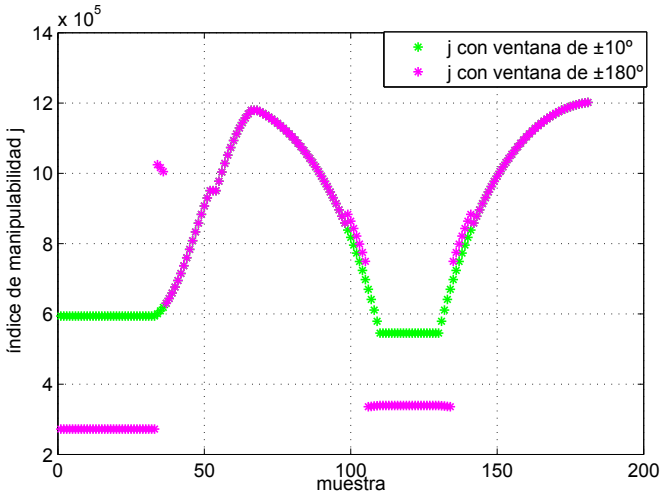


Figura 5.13. Corrección del índice de manipulabilidad optimizado

un control cinemático de impedancia basado en posición (Apéndice A) que se ejecuta en el controlador R30 iA Mate, del robot industrial FANUC LR Mate 200iC recién adquirido en el laboratorio.

Para lograr juntar ambos algoritmos, se realizaron algunas modificaciones al sistema teleoperado implementado anteriormente. El principal cambio fue que ahora toda la carga computacional del algoritmo de optimización es implementada dentro del controlador del robot. Anteriormente el controlador únicamente recibía la cadena de posiciones optimizadas a colocar el manipulador. Ahora, el controlador sensa

continuamente si existe detección de fuerzas en el efector final y, de acuerdo a las mediciones obtenidas, decide la acción del manipulador a implementar. En la Figura 5.14 se describe a grandes rasgos el algoritmo que se ejecuta en el controlador.

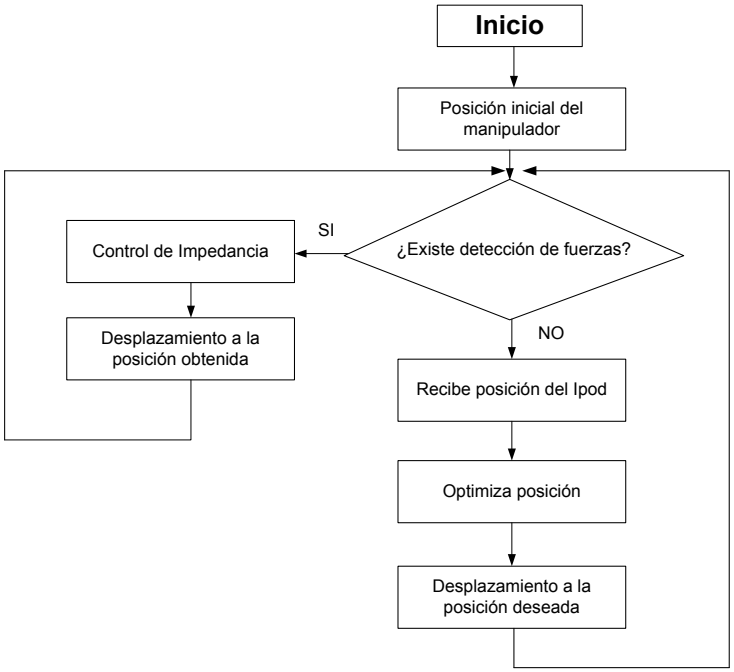


Figura 5.14. Algoritmo en el controlador del robot

Para la validación de este nuevo sistema teleoperado, se propusó el escenario de la Figura 5.15, en el cual el dispositivo maestro envía una trayectoria en línea recta. En el momento en el que el efector final del robot se encuentra con el obstáculo, éste debe modificar su trayectoria mediante el control de impedancia, para interactuar con el obstáculo tratando de seguir la trayectoria deseada. Cuando ya no se detecte la presencia de fuerzas de contacto, el manipulador esclavo debe regresar a su trayectoria original.

Para este experimento, se utilizó una rampa con superficie de unicel, como se observa en la Figura 5.15. La rampa fue arbitrariamente colocada dentro del espacio de trabajo del robot. Los resultados de la fase de seguimiento se muestran el la Figura 5.16, en color verde la trayectoria deseada en línea recta enviada desde el dispositivo maestro y en rosa, la trayectoria obtenida del robot. Se observa que el robot, al entrar en contacto con la rampa, adopta la geometría de la superficie con la cual interactua, y regresa a su trayectoria original al terminar la interacción. La forma de la línea en zigzag de la trayectoria del robot es debida a que el robot, al detectar un obstáculo en su trayectoria, mediante el control de impedancia calcula un pequeño desplazamiento en sentido contrario para evitar una colisión con el obstáculo. Sin embargo, como el dispositivo maestro continua enviando datos de la trayectoria en línea recta, el manipulador intenta retomar la trayectoria deseada, continuando con la interacción de la rampa. Se observa en la Figura 5.16 que, conforme la pendiente de la rampa aumenta, también aumenta la distancia calculada por el control de



Figura 5.15. Escenario para el control de impedancia

impedancia.

Durante la interacción con la rampa, las fuerzas de contacto son reguladas por los parámetros de impedancia, los cuáles fueron seleccionados como: $\mathbf{K}_d = diag\begin{pmatrix} 0.2 & 0.2 & 0.2 \end{pmatrix} N/m$, $\mathbf{B}_d = diag\begin{pmatrix} 0.1 & 0.1 & 0.1 \end{pmatrix} Ns/m$, $\mathbf{M}_d = diag\begin{pmatrix} 0.02 & 0.02 & 0.02 \end{pmatrix} kg$. Esta selección de parámetros se basó

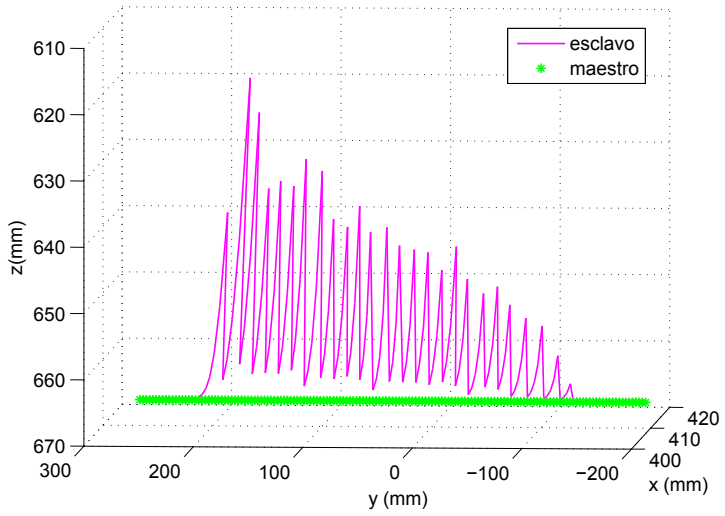


Figura 5.16. Trayectoria preplaneada del sistema teleoperado

en la respuesta del manipulador al interactuar con la rampa, pues con muchos valores, el robot no era capaz de terminar la trayectoria, ya que las fuerzas de contacto eran muy grandes y el manipulador no podía alcanzar la posición que obtenía del control de impedancia, o la respuesta del robot era muy lenta y las fuerzas de contacto muy pequeñas, las cuales no permitían al robot una interacción adecuada con la superficie de unicel.

La gráfica de las fuerzas de contacto obtenidas (Ver Figura 5.17), respecto a la trayectoria de la Figura 5.16, muestra como las fuerzas van aumentando conforme la inclinación de la rampa se incrementa. Al terminar la rampa, los valores de las fuerzas tienen regresan a cero, debido a que, ya no existe interacción con el entorno.

5.4.1. Variación de parámetros de impedancia

La interacción dinámica entre el manipulador y el entorno puede ser regulada modificando la sintonía de los parámetros de impedancia, de acuerdo con el comportamiento deseado. Por estas razones, se variaron los parámetros de impedancia en un rango de un valor 10 veces más grande y pequeño, comparados con los valores utilizados en la trayectoria de la Figura 5.16.

En primer lugar, se realizó una variación del parámetro de rigidez $\mathbf{K}_d$. Los resultados de variar este parámetro se observan en la Figura 5.18. Se puede observar en 5.18(a), que si se aumenta el valor del parámetro de rigidez a $\mathbf{K}_d = diag\begin{pmatrix} 2 & 2 & 2 \end{pmatrix} N/m$ el robot es incapaz de interactuar por completo con la rampa, se observa que en el último tramo de la trayectoria, cuando el robot interactúa con la parte más elevada de la rampa las fuerzas detectadas por el sensor aumentan, lo que provoca que el manipulador no alcance la

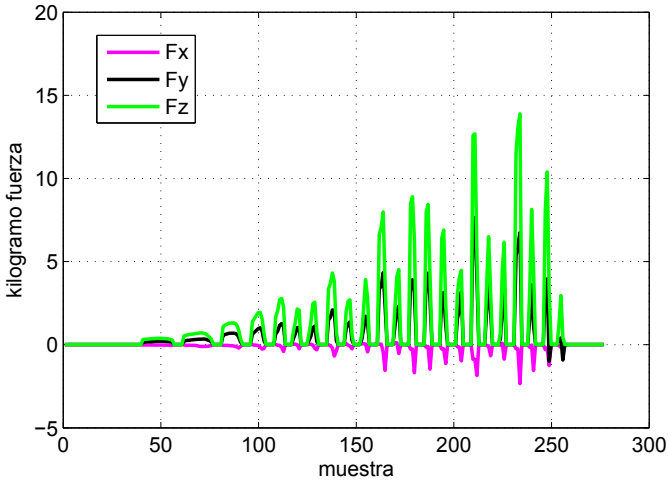
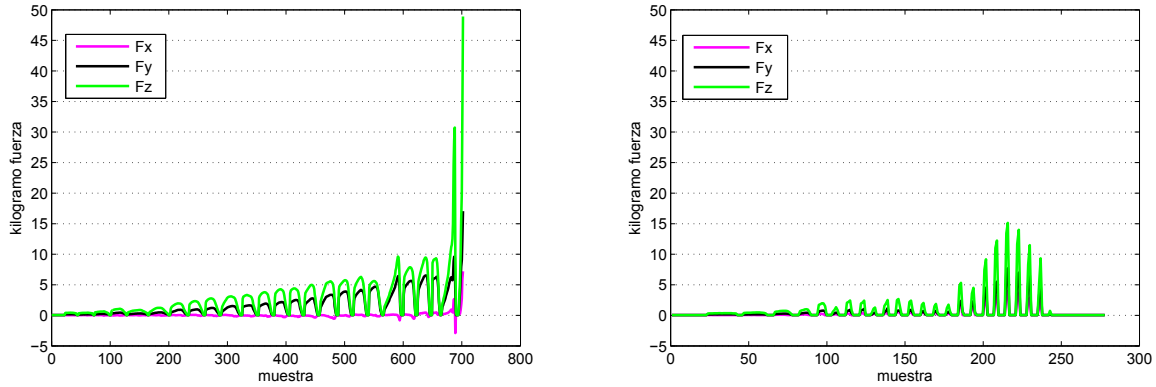


Figura 5.17. Valores de las fuerzas de contacto durante la interacción con la rampa

posición deseada que se obtiene del algoritmo del control de impedancia. Sin embargo, su velocidad aumenta debido a que, al momento de corregir su posición al detectar el contacto, el efector final se separa de la rampa una distancia pequeña, de aproximadamente 5cm. Por otro lado, en 5.18(b) se observa que, al disminuir el parámetro de rigidez a $\mathbf{K}_d = diag\begin{pmatrix} 0.02 & 0.02 & 0.02 \end{pmatrix} N/m$, el robot interactúa exitosamente con la rampa y regresa a su trayectoria original; su velocidad disminuye debido a que el manipulador, al recalcular la nueva posición, se aleja un poco más de la rampa.



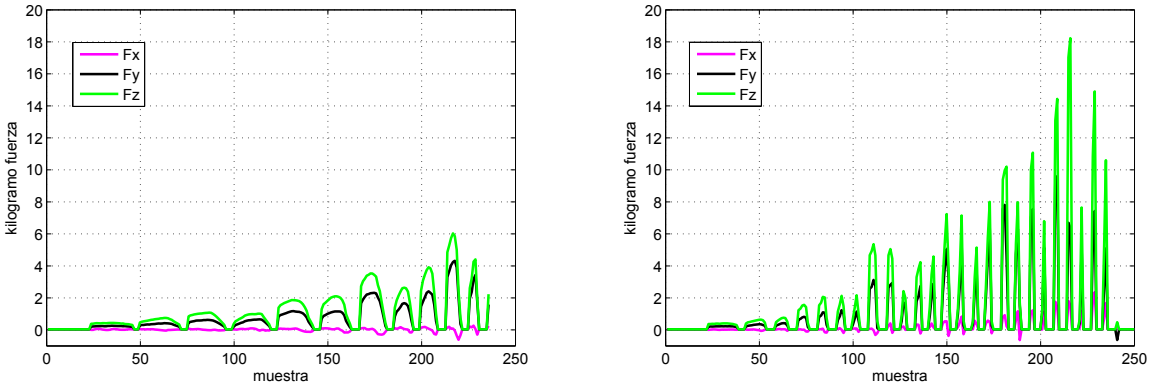
(a) Variación del parámetro de rigidez a $\mathbf{K}_d$ = 2

(b) Variación del parámetro de rigidez a $\mathbf{K}_d$ = 0.02

Figura 5.18. Variación del parámetro de rigidez $\mathbf{K}_d$

Posteriormente, se realizó una variación del parámetro de amortiguamiento $\mathbf{B}_d$. En la Figura 5.19(a) el parámetro $\mathbf{B}_d$ tiene un valor de $\mathbf{B}_d = diag\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$. Se puede observar que el comportamiento transitorio es más suave y los valores de las fuerzas son más pequeñas, sin embargo, el manipulador no logra terminar la trayectoria debido a que el robot marca un error de límite de junta. Si se disminuye el valor

del parámetro a $\mathbf{B}_d = diag\begin{pmatrix} 0.01 & 0.01 & 0.01 \end{pmatrix}$, el robot interactua exitosamente con la rampa, pero su comportamiento transitorio es más rápido, y los valores de sus fuerzas son aun más altos.

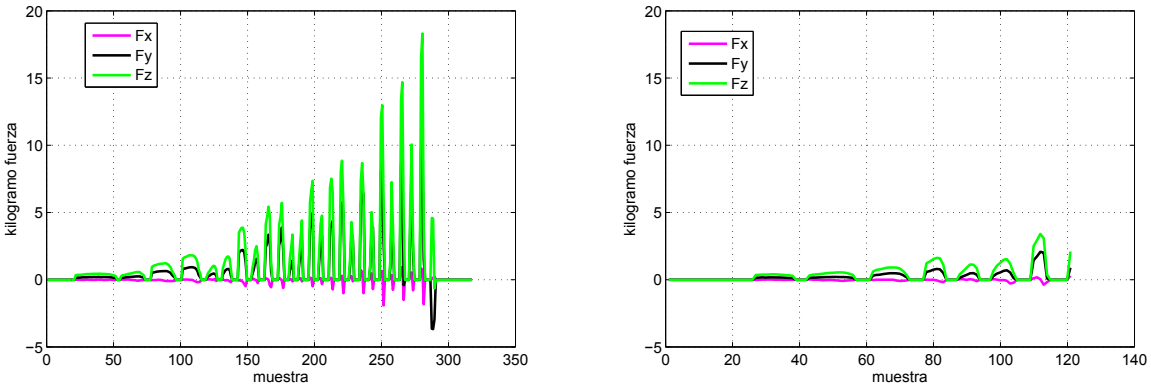


(a) Variación del parámetro de amortiguamiento a $\mathbf{B}_d$=1

(b) Variación del parámetro de amortiguamiento a $\mathbf{B}_d = 0.01$

Figura 5.19. Variación del parámetro de amortiguamiento $\mathbf{B}_d$

Finalmente, se varió el parámetro de inercia $\mathbf{M}_d$. En la Figura 5.20(a), al parámetro de inercia se le asignó un valor de $\mathbf{M}_d = diag\begin{pmatrix} 0.2 & 0.2 & 0.2 \end{pmatrix} kg$. El manipulador interactúa de forma exitosa la rampa, sin embargo, conforme va detectando fuerzas más grandes, su respuesta transitoria disminuye; esto provoca que se mueva de manera rápida, comparada con la gráfica 5.20(b), en la cual se asignó un valor de $\mathbf{M}_d = diag\begin{pmatrix} 0.002 & 0.002 & 0.002 \end{pmatrix} kg$. Se observa que la respuesta transitoria es lenta y los valores de las fuerzas pequeños. A pesar de esto, el manipulador no logró interactuar correctamente con la rampa debido a que el controlador señaló un error por haber alcanzado el límite de desplazamiento de la junta.



(a) Variación del parámetro de inercia a $\mathbf{M}_d = 0.2$

(b) Variación del parámetro de inercia a $\mathbf{M}_d = 0.002$

Figura 5.20. Variación del parámetro de inercia $\mathbf{M}_d$

Se puede concluir que la selección de los parámetro de impedancia es crítica al momento de establecer la interacción con fuerzas externas sobre una superficie, en este caso, los valores donde el robot interactuó

correctamente con la rampa, fueron aquellos en los cuales la respuesta transitoria no era tan lenta, y las fuerzas de contacto oscilaban entre los valores de 5 y 10 kilogramo fuerza en la parte más alta de la rampa.

Conclusiones y trabajo futuro

Durante el desarrollo de este trabajo de tesis se presentó el diseño e implementación de un sistema teleoperado unilateralmente, que optimiza la posición de un robot industrial al momento de seguir una trayectoria definida mediante un dispositivo maestro. La optimización de la posición del robot se hace mediante la maximización del índice de manipulabilidad. Los objetivos planteados al inicio del trabajo han sido alcanzados, dentro de los cuales se pueden mencionar los siguientes:

Se ha implementado con éxito, de forma inalámbrica, la teleoperación entre un dispositivo maestro, en este caso un Ipod Touch y dos robots industriales el FANUC el M16-iB20T y el LRMate 200iC, cumpliendo con el objetivo de diseñar el sistema de forma general para que sea adaptable a diversos manipuladores industriales.

Se logró diseñar una interfaz de usuario bajo los conceptos de computación ubicua, utilizando las herramientas que brinda el dispositivo móvil, de manera que es intuitiva y ergonómica para el control de posición y orientación de un robot industrial.

Se corrigió el problema de la asimetría en el sistema teleoperado, mediante un control cinemático, el cual asigna de forma automática el valor de la última variable de control, de forma que la posición del robot sea optimizada.

Se logró implementar un algoritmo de optimización de la posición del manipulador al seguir una trayectoria. Este algoritmo se implementó tanto en el dispositivo móvil como en el controlador del robot. Este último es importante ya que, gracias a las instrucciones de alto nivel con las que cuenta el lenguaje de programación KAREL, el algoritmo de optimización puede adaptarse a cualquier controlador FANUC, únicamente es necesario cambiar el jacobiano de cada manipulador.

Finalmente, se logró la integración de un algoritmo de control de impedancia dentro del sistema teleoperado, el cual permite modular las fuerzas de interacción entre el robot y su entorno. La variación de los parámetros de impedancia permiten ajustar al sistema de acuerdo a la rigidez del entorno para que el efector final del

robot se adapte a la forma de la superficie con la que interactúa.

Trabajo Futuro

Con base en la experiencia adquirida durante el desarrollo del trabajo de investigación de esta tesis, surgen algunas áreas de oportunidad:

- Mejorar la comunicación con el robot, esto con el fin de hacer más rápida la tarea de seguimiento de las trayectorias. Una opción es la implementación de un buffer circular de datos entre el robot y el dispositivo móvil.
- La implementación del algortimo de optimización del manipulador en una tarea industrial actual y recurrente.
- Otra área importante a tomar en cuenta es la correcta elección de los parámetros de impedancia, los cuáles influyen de manera importante en el comportamiento del manipulador al interactuar con fuerzas externas. Una opción es desarrollar un método de sintonización de parámetros con base en diferentes materiales de contacto.

Apéndice A

Control de impedancia

En la presente sección se describen una serie de aspectos relacionados con el uso de un modelo de control de impedancia. Hogan [32] propone un modelo de interacción que consiste en un modelo generalizado de masa, resorte, amortiguador, donde la relación entre la trayectoria del efector final x, la trayectoria deseada x_d y el vector de fuerzas de contacto con el entorno f_e, está dada por la siguiente expresión:

$$\mathbf{f}_e = \mathbf{K}_d(x_d - x) + \mathbf{B}_d(\dot{x}_d - \dot{x}) + \mathbf{M}_d(\ddot{x}_d - \ddot{x}) \tag{A.1}$$

donde $\mathbf{K}_d \in \Re^{mxm}$ es la matriz de rigidez, $\mathbf{B}_d \in \Re^{mxm}$ es la matriz de amortiguamiento y $\mathbf{M}_d \in \Re^{mxm}$ es la matriz de inercia. $\mathbf{f}_e \in \Re^m$ denota el vector de fuerzas de contacto o interacción. Estos parámetros caracterizan la impedancia $\mathbf{Z}_d \in \Re^{mxm}$ de la relación A.1, la cual es dada en el dominio de la frecuencia por

$$\mathbf{Z}_d(s) = \mathbf{K}_d + \mathbf{B}_d s + \mathbf{M}_d s^2 \tag{A.2}$$

Estos parámetros de impedancia pueden seleccionarse para corresponder a varias tareas de manipulación. Típicamente, alta rigidez es especificada en direcciones donde el entorno es dócil y es importante un posicionamiento preciso. Baja rigidez es especificada en direcciones donde el entorno es rígido, o cuando deben mantenerse fuerzas de interacción pequeñas. Grandes valores de $\mathbf{B}_d$ son especificados cuando debe disiparse energía, y los valores de $\mathbf{M}_d$ pueden ser utilizados para mejorar la respuesta del efector final debida al contacto externo.

A.1. Control de Impedancia Basado en Posición

Con el propósito de implementar un controlador de impedancia en robots manipuladores industriales, sin modificar su arquitectura cerrada, se implementaron los algoritmos de control cinemático de impedancia.

Lawrence [31] presenta un enfoque de control de impedancia basado en posición, donde las fuerzas y los pares son explícitamente medidos usando un sensor de fuerza/par ubicado en la muñeca del manipulador. Con base en las fuerzas y pares de contacto, la posición del robot se ajusta y la nueva configuración se envía al lazo de control interno. En particular, se crea un vector de ajuste de posición donde $x_a \in \Re^m$ mediante el filtrado de las fuerzas de interacción medidas para satisfacer

$$f_e = \mathbf{K}_d x_a + \mathbf{B}_d \dot{x}_a + \mathbf{M}_d \ddot{x}_a \tag{A.3}$$

Al tomar x_a como un vector de ajuste de posición y obtener su transformada de Laplace, se puede ver que :

$$x_a(s) = (\mathbf{K}_d + \mathbf{B}_d s + \mathbf{M}_d s^2)^{-1} f_e(s) \tag{A.4}$$

Con la simplificación de que $\mathbf{K}_d$, $\mathbf{B}_d$ y $\mathbf{M}_d$ sean diagonales, esto se reduce a un filtro pasa bajas de cada componente de la fuerza f_e para generar cada componente de x_a. El ajuste x_a es restado a la trayectoria de posición deseada para generar

$$x_c = x_d - x_a \tag{A.5}$$

Se puede observar que, cuando no existe contacto con el entorno, $f_e = 0$, entonces x_c = x_d. Si al controlar el manipulador se logra el correcto posicionamiento, es decir x = x_c, entonces

$$x_a = x_d - x \tag{A.6}$$

y por ende A.3 y A.4 satisfacen la impedancia deseada especificada por A.1 y A.2. Este enfoque permite un control de posición preciso dentro del lazo de control de impedancia, haciéndolo atractivo para su uso en manipuladores industriales, debido a que están diseñados como dispositivos de posicionamiento preciso (o al menos repetitivo). Sin embargo, sufre la incapacidad de proveer impedancias muy suaves, es decir, rigidez y amortiguamiento pequeños. Generalmente sería deseable en casos donde se requiere control de posición articular rígido, es decir un control de posición con mayor seguridad o un correcto posicionamiento en algunas direcciones cartesianas.

Apéndice B

Fotografías

En esta sección se exponen una serie de fotografías que ilustran los dispositivos utilizados para este proyecto, que se encuentran en el Laboratorio de Robótica de la Facultad de Ingeniería en la UASLP.

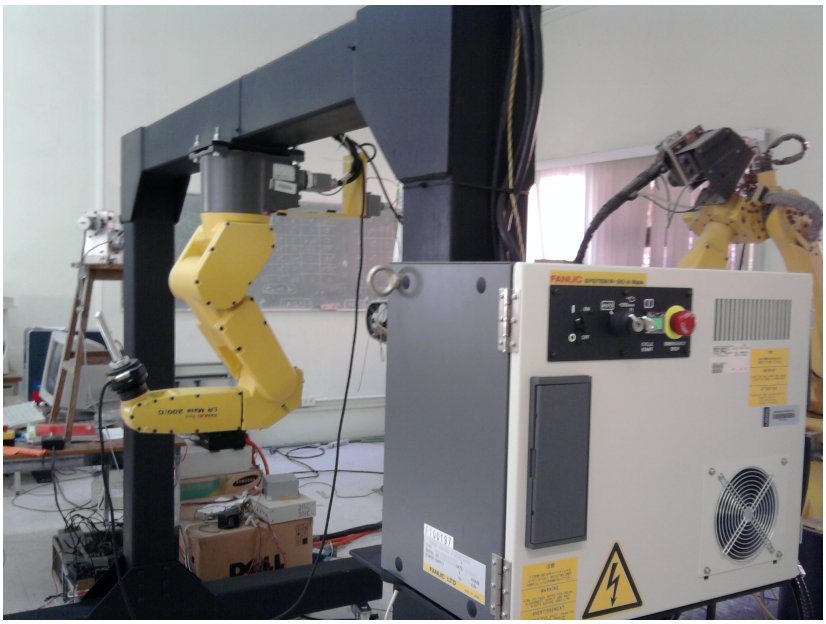


Figura B.1. Robot Fanuc Fanuc LR Mate 200iC



Figura B.2. Robot Fanuc M16 iB 20T



Figura B.3. Unidad de control del Fanuc M16 iB 20T

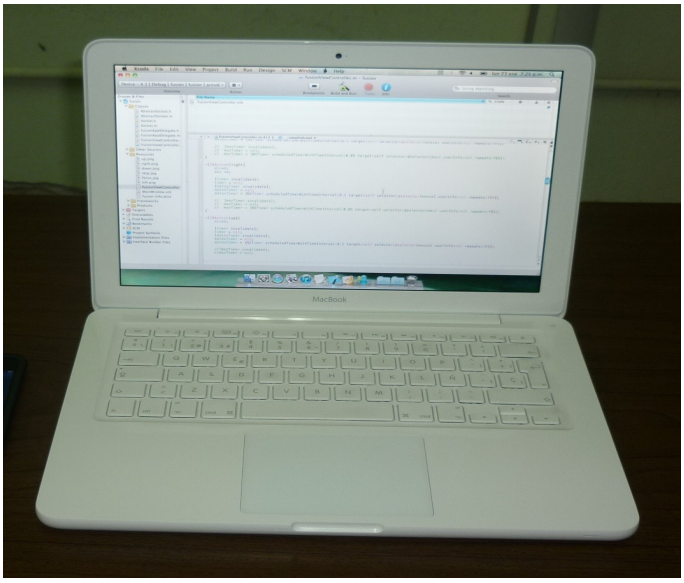


Figura B.4. MacBook utilizada para la programación del dispositivo móvil



Figura B.5. Dispositivo móvil Ipod Touch utilizado para el sistema teleoperado

Apéndice C

Códigos de Programa

La programación de los algoritmos necesarios para implementar el sistema teleoperado se describen a continuación. El primer programa es el algoritmo que corre dentro del controlador del robot, esta escrito el lenguaje KAREL, propia de la marca FANUC.

```
1 --------------------------------------------------------------------------------
2 ----- PROGRAMA PARA MOVER EL ROBOT EN ESPACIO CARTESIANO EN LENGUAJE KAREL------
3 --------------------------------------------------------------------------------
4 PROGRAM tesis
5 VAR
6           file_var   : FILE
7           entry      : INTEGER
8           Dato       : STRING[80]
9           STATUS     : INTEGER
10          set_joint  : ARRAY[6] OF REAL
11          juntas     : ARRAY[10] OF REAL
12          p_joint    : JOINTPOS6
13          graf       : FILE
14          p_xyz      : XYZWPR
15          clock       :INTEGER
16
17 ROUTINE RecibeJ
18 VAR
19   indx         : INTEGER
20   chr_cnt      : INTEGER
21   segment      : INTEGER
22   chr_strt     : INTEGER
23   StringSize   : INTEGER
24   i            : INTEGER
25   nSemi        : INTEGER
```

```
BEGIN
  MSG_DISCO('S3:',STATUS)
  MSG_CONNECT('S3:',STATUS)
  IF STATUS = 0 THEN
      CLOSE FILE file_var
      OPEN FILE file_var ('RW','S3:')
          STATUS = IO_STATUS(file_var)
          IF STATUS = 0 THEN
          READ file_var(Dato::51)
          nSemi=0
          i=1
          IF SUB_STR(Dato, i,1)<>';' THEN
          RETURN
          ENDIF
          StringSize =STR_LEN(Dato);
          REPEAT
          IF SUB_STR(Dato, i,1)=';' THEN
          nSemi = nSemi + 1
          ENDIF
          i = i + 1
          IF i>StringSize  THEN
          RETURN
          ENDIF
          UNTIL SUB_STR(Dato, i,1)='!'
          IF nSemi<>7 THEN
          RETURN
          ENDIF
          indx=2
          segment = 1
          REPEAT
          chr_cnt=0
          REPEAT
          IF SUB_STR(Dato,indx,1)<>';' THEN
          chr_cnt = chr_cnt + 1
          indx = indx + 1
          ENDIF
          UNTIL SUB_STR(Dato,indx,1)=';'
          IF SUB_STR(Dato,indx,1)=';' THEN
          indx = indx + 1
          chr_strt=indx-chr_cnt-1
          CNV_STR_REAL(SUB_STR(Dato,chr_strt,chr_cnt),juntas[segment])
          ENDIF
          segment = segment + 1
          UNTIL SUB_STR(Dato,indx,1)='!'
          set_joint[1] = juntas[1] -1000
          set_joint[2] = juntas[2] -360
          set_joint[3] = juntas[3] -360
```

```
          set_joint[4] = juntas[4] -360
          set_joint[5] = juntas[5] -360
          set_joint[6] = juntas[6] -360
          ENDIF
  ENDIF
    MSG_DISCO('S3:',STATUS)
    CLOSE FILE file_var

END RecibeJ
-----------------------------------------------------------------------------
BEGIN
   $GROUP[1].$SPEED = $PARAM_GROUP[1].$SPEEDLIMJNT
   $GROUP[1].$UTOOL = $MNUTOOL[1,1]
   $GROUP[1].$USE_TURNS = FALSE
   $GROUP[1 ].$USEMAXACCEL=TRUE$
    SET_FILE_ATR(file_var, ATR_IA)
    SET_FILE_ATR(graf, ATR_IA)
    OPEN FILE graf ('RW', 'tray.kl')
    SET_VAR(entry, '*SYSTEM*','$HOSTS_CFG[3].$SERVER_PORT',59002,STATUS)
   aqui::
       CONNECT TIMER TO clock
       RecibeJ
       CNV_REL_JPOS(set_joint, p_joint, STATUS)
       MOVE TO p_joint
       p_joint = CURPOS(0,0,1)
       p_xyz = p_joint
       WRITE graf(clock)
       WRITE graf(p_xyz.x)
       WRITE graf(p_xyz.y)
       WRITE graf(p_xyz.z)
       WRITE graf(p_xyz.w)
       WRITE graf(p_xyz.p)
       WRITE graf(p_xyz.r,CR)
       clock=0
   GOTO aqui

CLOSE FILE graf
END tesis
```

El algoritmo que corre dentro del dispositivo móvil, esta escrito en lenguaje Objetive C, el cuál descompone el programa en dos partes: el encabezado donde se declaran las variables globales, herramientas y acciones a utilizar. La segunda parte es la que describe las funciones a realizar en cada acción declarada en el encabezado.

PRIMERA PARTE

```
1 //  tesisViewController.h
2 #define rad    (3.1416f/180.0f)
3 #define grados (180.0f/3.1416f)
4 #import <UIKit/UIKit.h>
5 #import <CoreFoundation/CoreFoundation.h>
6 #import "Socket.h"
7
8 @interface tesisViewController : UIViewController <UIAccelerometerDelegate> {
9   //IBOutlet UIImageView *image;
10  //NSTimer *movTimer;
11  NSTimer *datosTimer;
12  int dir;
13  int mov;
14  BOOL isConnected;
15  AbstractSocket* socket;
16  IBOutlet UILabel *label;
17  IBOutlet UILabel *label2;
18  int pos;
19  UIAccelerometer *acc;
20  NSTimer *timer;
21
22  //datos ctes. robot
23  float l1;
24  float l2;
25  float l3;
26  float l4;
27  float ex;
28  float ey;
29  float ez;
30  // variables de junta del robot
31  float d1;
32  float t2;
33  float t3;
34  float t4;
35  float t5;
36  float t6;
37  //variables cartesianas del robot
38  float x;
39  float y;
40  float z;
41  float w;
42  float p;
43  float r;
44  // variables del método numérico
45  float eopt;
46  float er;
47  float f1;
```

```
48  float f2;
49  float eo;
50  float eta;
51  float xl;
52  float yl;
53  float zl;}
54
55 -(IBAction)up;
56 -(IBAction)down;
57 -(IBAction)left;
58 -(IBAction)right;
59 -(IBAction)stop;
60 -(IBAction)upZ;
61 -(IBAction)downZ;
62 -(void)envio;
63 -(void)inclinacion;
64 -(IBAction)salir;
65 -(IBAction)inicio;
66
67 @end
```

SEGUNDA PARTE

```
1 tesisViewController.m
2 #import "tesisViewController.h"
3 @implementation tesisViewController
4 // se detecta la inclinación del dispositivo por medio de los acelerómetros
5 -(void)accelerometer:(UIAccelerometer *)acc didAccelerate:(UIAcceleration *)
      acceleration {
6   if (acceleration.x<0.6 && acceleration.x>-0.6  ) {
7     pos = 0; }
8   if (acceleration.y>0.6) {
9     pos = 1;}
10  if (acceleration.y<-0.6) {
11    pos = 2; }
12  if (acceleration.x<-0.6) {
13    pos = 3;}
14  if (acceleration.x>0.6 ) {
15    pos = 4;}}
16 // se detecta el contacto de dos dedos en la pantalla para cambiar a modo
      orientación
17 - (void) touchesBegan:(NSSet *) touches withEvent:(UIEvent *) event
18 {   NSSet *allTouches = [event allTouches];
19  switch ([allTouches count]) {
20    case 1:
21    {}break;
22    case 2:
```

```
23    { [datosTimer invalidate];
24      datosTimer = nil;
25      label2.text=@"mandando orientación";
26      [timer invalidate];
27      timer = nil;
28      timer = [NSTimer scheduledTimerWithTimeInterval:0.05 target:self selector:
            @selector(inclinacion) userInfo:nil repeats:YES];
29      } break;}}
30// se detecta la dirección del eje deseado para la posición del robot
31-(IBAction)up{
32  dir=1;
33  mov =1;
34  [timer invalidate];
35  timer = nil;
36  [datosTimer invalidate];
37  datosTimer = nil;
38  datosTimer = [NSTimer scheduledTimerWithTimeInterval:0.05 target:self selector:
        @selector(envio) userInfo:nil repeats:YES];}
39-(IBAction)down{
40  dir=2;
41  mov =2;
42  [timer invalidate];
43  timer = nil;
44  [datosTimer invalidate];
45  datosTimer = nil;
46  datosTimer = [NSTimer scheduledTimerWithTimeInterval:0.05 target:self selector:
        @selector(envio) userInfo:nil repeats:YES];}
47-(IBAction)left{
48  dir=3;
49  mov =3;
50  [timer invalidate];
51  timer = nil;
52  [datosTimer invalidate];
53  datosTimer = nil;
54  datosTimer = [NSTimer scheduledTimerWithTimeInterval:0.05 target:self selector:
        @selector(envio) userInfo:nil repeats:YES];}
55-(IBAction)right{
56  dir=4;
57  mov =4;
58  [timer invalidate];
59  timer = nil;
60  [datosTimer invalidate];
61  datosTimer = nil;
62  datosTimer = [NSTimer scheduledTimerWithTimeInterval:0.05 target:self selector:
        @selector(envio) userInfo:nil repeats:YES];}
63-(IBAction)upZ{
64  dir=5;
```

```
65  [timer invalidate];
66  timer = nil;
67  [datosTimer invalidate];
68  datosTimer = nil;
69  datosTimer = [NSTimer scheduledTimerWithTimeInterval:0.05 target:self selector:
        @selector(envio) userInfo:nil repeats:YES];}
70 -(IBAction)downZ{
71  dir=6;
72  [timer invalidate];
73  timer = nil;
74  [datosTimer invalidate];
75  datosTimer = nil;
76  datosTimer = [NSTimer scheduledTimerWithTimeInterval:0.05 target:self selector:
        @selector(envio) userInfo:nil repeats:YES];}
77 // se forma la cadena de datos a enviar al robot y se optimiza la posición
78 -(void)envio
79 { switch (dir) {
80    case 1:{
81      if (x > (xl-500)) {x=x-5;}
82      else    {x=x;}
83    }break;
84    case 2:{
85      if (x < (xl+500)) {x=x+5;}
86      else    {x=x;}
87    }break;
88    case 3:{
89      if (y > (yl-1000)) {y=y-5;}
90      else    {y=y;}
91    }break;
92    case 4:{
93      if (y < (yl+1000)) {y=y+5;}
94      else    {y=y;}
95    }break;
96    case 5:{
97      if (z < (zl-500)) {z=z;}
98      else    {z=z-5;}
99    }break;
100   case 6:{
101     if (z > (zl+500)) {z=z;}
102     else    {z=z+5;}
103   }break;
104   default:
105   {}break;}
106
107 w=(w)*rad;
108 p=(p)*rad;
109 r=(r)*rad;
```

```
110  float ei;
111  float ef;
112  if (eopt < -180) {ei=-180;}
113  else {ei = eopt - (10*rad);}
114  if (eopt > 180) {ef = 180;}
115  else {ef = eopt + (10*rad);}
116  float sd = (sqrt(5)-1)/2;
117  float d = sd*(ef-ei);
118  float iter=1;
119  float e1 = ei+d;
120  float e2 = ef-d;
121  float es = 0.001;
122  float ea;
123  float beta;
124
125  float beta1;
126  float xw = x - ex*(cos(p)*cos(r)) - ey*(cos(r)*sin(w)*sin(p)-cos(w)*sin(r)) - ez
         *(sin(w)*sin(r)+cos(w)*cos(r)*sin(p)) - l4*(sin(w)*sin(r)+cos(w)*cos(r)*sin(p)
         );
127  float yw = y - ex*(cos(p)*sin(r)) - ey*(cos(w)*cos(r)+sin(w)*sin(p)*sin(r)) - ez
         *(cos(w)*sin(p)*sin(r)-cos(r)*sin(w)) - l4*(cos(w)*sin(p)*sin(r)-cos(r)*sin(w)
         );
128  float zw = z + ex*(-sin(p)) + ey*(cos(p)*sin(w)) + ez*(cos(w)*cos(p)) + l4*(cos(w
         )*cos(p));
129  d1 = yw;
130  float R = sqrt(pow(xw,2)+pow(zw,2));
131  float P = sqrt(pow(l3,2)+pow(l2,2));
132  float alfa = atan2(zw,xw);
133  float D = (-pow(P,2) + pow(R,2) + pow(l1,2))/(2*R*l1);
134  if (D>1) {beta1=0;}
135  else {beta1 = atan2(sqrt(1-pow(D,2)),D);}
136  t2= (alfa+beta1);
137  float z3=l1*sin(t2);
138  float x3=l1*cos(t2);
139  float gama = atan2(l3,l2);
140  float delta = atan2(zw-z3,xw-x3);
141  t3 =(gama - delta);
142  t4 = (-atan2((cos(r)*sin(w)-cos(w)*sin(p)*sin(r)),(cos(t3)*(sin(w)*sin(r)+cos(w)*
         cos(r)*sin(p))+cos(w)*cos(p)*sin(t3))));
143  float R03 = sin(t3)*(sin(w)*sin(r)+cos(w)*cos(r)*sin(p))-cos(w)*cos(p)*cos(t3);
144  t5 = (atan2(sqrt(1- pow(R03,2)),R03));
145  t6 = (atan2(-(-sin(t3)*(cos(w)*sin(r)-cos(r)*sin(w)*sin(p))-cos(p)*cos(t3)*sin(w)
         ),-(cos(t3)*sin(p)+cos(p)*cos(r)*sin(t3))));
146 float j = cálculo el índice de manipulabilidad
147
148  eta = e1 ;
```

```
149 xw = x + ey*(cos(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w)) - sin(eta)*(sin(r)*
        sin(w) + cos(r)*cos(w)*sin(p))) - ez*(cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*
        sin(p)) + sin(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w))) - l4*(cos(eta)*(sin
        (r)*sin(w) + cos(r)*cos(w)*sin(p)) + sin(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*
        sin(w))) - ex*cos(p)*cos(r);
150 yw = y - ey*(cos(eta)*(cos(r)*cos(w) + sin(p)*sin(r)*sin(w)) - sin(eta)*(cos(r)*
        sin(w) - cos(w)*sin(p)*sin(r))) + ez*(cos(eta)*(cos(r)*sin(w) - cos(w)*sin(p)*
        sin(r)) + sin(eta)*(cos(r)*cos(w) + sin(p)*sin(r)*sin(w))) + l4*(cos(eta)*(cos
        (r)*sin(w) - cos(w)*sin(p)*sin(r)) + sin(eta)*(cos(r)*cos(w) + sin(p)*sin(r)*
        sin(w))) - ex*cos(p)*sin(r) ;
151 zw = z - ex*sin(p) + ey*(cos(eta)*cos(p)*sin(w) + cos(p)*sin(eta)*cos(w)) + ez*(
        cos(eta)*cos(p)*cos(w) - cos(p)*sin(eta)*sin(w)) + l4*(cos(eta)*cos(p)*cos(w)
        - cos(p)*sin(eta)*sin(w));
152 d1 = yw;
153 R = sqrt(pow(xw,2)+pow(zw,2));
154 P = sqrt(pow(l3,2)+pow(l2,2));
155 alfa = atan2(zw,xw);
156 D = (-pow(P,2) + pow(R,2) + pow(l1,2))/(2*R*l1);
157 if (D>1) {beta=0;}
158 else {beta = atan2(sqrt(1-pow(D,2)),D);}
159 t2= (alfa+beta);
160 z3=l1*sin(t2);
161 x3=l1*cos(t2);
162 gama = atan2(l3,l2);
163 delta = atan2(zw-z3,xw-x3);
164 t3 =(gama - delta);
165 float R33 = sin(t3)*(cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p)) + sin(eta)*(
        cos(w)*sin(r) - cos(r)*sin(p)*sin(w))) - cos(t3)*(cos(eta)*cos(p)*cos(w) - cos
        (p)*sin(eta)*sin(w));
166 t4=-atan2(cos(eta)*(cos(r)*sin(w) - cos(w)*sin(p)*sin(r)) + sin(eta)*(cos(r)*cos(
        w) + sin(p)*sin(r)*sin(w)),(sin(t3)*(cos(eta)*cos(p)*cos(w) - cos(p)*sin(eta)*
        sin(w)) + cos(t3)*(cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p)) + sin(eta)
        *(cos(w)*sin(r) - cos(r)*sin(p)*sin(w))));
167 t5=(atan2(sqrt(1- pow(R33,2)),R33));
168 t6=atan2(-(- sin(t3)*(cos(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w)) - sin(eta)
        *(sin(r)*sin(w) + cos(r)*cos(w)*sin(p))) - cos(t3)*(cos(eta)*cos(p)*sin(w) +
        cos(p)*sin(eta)*cos(w))),  -(cos(t3)*sin(p)+cos(p)*cos(r)*sin(t3)));
169 f1 = cálculo el índice de manipulabilidad
170
171 eta = e2 ;
172 xw = x + ey*(cos(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w)) - sin(eta)*(sin(r)*
        sin(w) + cos(r)*cos(w)*sin(p))) - ez*(cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*
        sin(p)) + sin(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w))) - l4*(cos(eta)*(sin
        (r)*sin(w) + cos(r)*cos(w)*sin(p)) + sin(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*
        sin(w))) - ex*cos(p)*cos(r);
173 yw = y - ey*(cos(eta)*(cos(r)*cos(w) + sin(p)*sin(r)*sin(w)) - sin(eta)*(cos(r)*
        sin(w) - cos(w)*sin(p)*sin(r))) + ez*(cos(eta)*(cos(r)*sin(w) - cos(w)*sin(p)*
```

```
        sin(r)) + sin(eta)*(cos(r)*cos(w) + sin(p)*sin(r)*sin(w))) + l4*(cos(eta)*(cos
        (r)*sin(w) - cos(w)*sin(p)*sin(r)) + sin(eta)*(cos(r)*cos(w) + sin(p)*sin(r)*
        sin(w))) - ex*cos(p)*sin(r) ;
174   zw = z - ex*sin(p) + ey*(cos(eta)*cos(p)*sin(w) + cos(p)*sin(eta)*cos(w)) + ez*(
        cos(eta)*cos(p)*cos(w) - cos(p)*sin(eta)*sin(w)) + l4*(cos(eta)*cos(p)*cos(w)
        - cos(p)*sin(eta)*sin(w));
175   d1 = yw;
176   R = sqrt(pow(xw,2)+pow(zw,2));
177   P = sqrt(pow(l3,2)+pow(l2,2));
178   alfa = atan2(zw,xw);
179   D = (-pow(P,2) + pow(R,2) + pow(l1,2))/(2*R*l1);
180   if (D>1) {beta=0;}
181   else {    beta = atan2(sqrt(1-pow(D,2)),D); }
182   t2= (alfa+beta);
183   z3=l1*sin(t2);
184   x3=l1*cos(t2);
185   gama = atan2(l3,l2);
186   delta = atan2(zw-z3,xw-x3);
187   t3 =(gama - delta);
188   R33 = sin(t3)*(cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p)) + sin(eta)*(cos(w)
        *sin(r) - cos(r)*sin(p)*sin(w))) - cos(t3)*(cos(eta)*cos(p)*cos(w) - cos(p)*
        sin(eta)*sin(w));
189   t4=-atan2(cos(eta)*(cos(r)*sin(w) - cos(w)*sin(p)*sin(r)) + sin(eta)*(cos(r)*cos(
        w) + sin(p)*sin(r)*sin(w)),(sin(t3)*(cos(eta)*cos(p)*cos(w) - cos(p)*sin(eta)*
        sin(w)) + cos(t3)*(cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p)) + sin(eta)
        *(cos(w)*sin(r) - cos(r)*sin(p)*sin(w)))));
190   t5=(atan2(sqrt(1- pow(R33,2)),R33));
191   t6=atan2(-(- sin(t3)*(cos(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w)) - sin(eta)
        *(sin(r)*sin(w) + cos(r)*cos(w)*sin(p))) - cos(t3)*(cos(eta)*cos(p)*sin(w) +
        cos(p)*sin(eta)*cos(w))),  -(cos(t3)*sin(p)+cos(p)*cos(r)*sin(t3)));
192   f2 = cálculo del índice de manipulabilidad
193
194   if (f1>f2)
195     {eopt = e1;er=f1; }
196   else {eopt = e2;er=f2;  }
197
198   do {
199     d=sd*d;
200     if (f1>f2)
201     { ei=e2;
202       e2=e1;
203       e1=ei+d;
204       f2=f1;
205
206       eta = e1;
207       xw = x + ey*(cos(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w)) - sin(eta)*(sin(
              r)*sin(w) + cos(r)*cos(w)*sin(p))) - ez*(cos(eta)*(sin(r)*sin(w) + cos(r)*
```

```
        cos(w)*sin(p)) + sin(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w))) - l4*(
        cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p)) + sin(eta)*(cos(w)*sin(r)
        - cos(r)*sin(p)*sin(w))) - ex*cos(p)*cos(r);
208     yw = y - ey*(cos(eta)*(cos(r)*cos(w) + sin(p)*sin(r)*sin(w)) - sin(eta)*(cos(
        r)*sin(w) - cos(w)*sin(p)*sin(r))) + ez*(cos(eta)*(cos(r)*sin(w) - cos(w)*
        sin(p)*sin(r)) + sin(eta)*(cos(r)*cos(w) + sin(p)*sin(r)*sin(w))) + l4*(
        cos(eta)*(cos(r)*sin(w) - cos(w)*sin(p)*sin(r)) + sin(eta)*(cos(r)*cos(w)
        + sin(p)*sin(r)*sin(w))) - ex*cos(p)*sin(r) ;
209     zw = z - ex*sin(p) + ey*(cos(eta)*cos(p)*sin(w) + cos(p)*sin(eta)*cos(w)) +
        ez*(cos(eta)*cos(p)*cos(w) - cos(p)*sin(eta)*sin(w)) + l4*(cos(eta)*cos(p)
        *cos(w) - cos(p)*sin(eta)*sin(w));
210     d1 = yw;
211     R = sqrt(pow(xw,2)+pow(zw,2));
212     P = sqrt(pow(l3,2)+pow(l2,2));
213     alfa = atan2(zw,xw);
214     D = (-pow(P,2) + pow(R,2) + pow(l1,2))/(2*R*l1);
215     if (D>1) {beta=0;}
216     else {    beta = atan2(sqrt(1-pow(D,2)),D); }
217     t2= (alfa+beta);
218     z3=l1*sin(t2);
219     x3=l1*cos(t2);
220     gama = atan2(l3,l2);
221     delta = atan2(zw-z3,xw-x3);
222     t3 =(gama - delta);
223     R33 = sin(t3)*(cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p)) + sin(eta)*(
        cos(w)*sin(r) - cos(r)*sin(p)*sin(w))) - cos(t3)*(cos(eta)*cos(p)*cos(w) -
         cos(p)*sin(eta)*sin(w));
224     t4=-atan2(cos(eta)*(cos(r)*sin(w) - cos(w)*sin(p)*sin(r)) + sin(eta)*(cos(r)*
        cos(w) + sin(p)*sin(r)*sin(w)),(sin(t3)*(cos(eta)*cos(p)*cos(w) - cos(p)*
        sin(eta)*sin(w)) + cos(t3)*(cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p)
        ) + sin(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w))));
225     t5=(atan2(sqrt(1- pow(R33,2)),R33));
226     t6=atan2(-(- sin(t3)*(cos(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w)) - sin(
        eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p))) - cos(t3)*(cos(eta)*cos(p)*
        sin(w) + cos(p)*sin(eta)*cos(w))),  -(cos(t3)*sin(p)+cos(p)*cos(r)*sin(t3)
        ));
227     f1 = cálculo del índice de manipulabilidad
228   }
229   else
230   {
231     ef=e1;
232     e1=e2;
233     e2=ef-d;
234     f1=f2;
235     eta = e2 ;
236     xw = x + ey*(cos(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w)) - sin(eta)*(sin(
        r)*sin(w) + cos(r)*cos(w)*sin(p))) - ez*(cos(eta)*(sin(r)*sin(w) + cos(r)*
```

```
            cos(w)*sin(p)) + sin(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w))) - l4*(
            cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p)) + sin(eta)*(cos(w)*sin(r)
            - cos(r)*sin(p)*sin(w))) - ex*cos(p)*cos(r);
237         yw = y - ey*(cos(eta)*(cos(r)*cos(w) + sin(p)*sin(r)*sin(w)) - sin(eta)*(cos(
            r)*sin(w) - cos(w)*sin(p)*sin(r))) + ez*(cos(eta)*(cos(r)*sin(w) - cos(w)*
            sin(p)*sin(r)) + sin(eta)*(cos(r)*cos(w) + sin(p)*sin(r)*sin(w))) + l4*(
            cos(eta)*(cos(r)*sin(w) - cos(w)*sin(p)*sin(r)) + sin(eta)*(cos(r)*cos(w)
            + sin(p)*sin(r)*sin(w))) - ex*cos(p)*sin(r) ;
238         zw = z - ex*sin(p) + ey*(cos(eta)*cos(p)*sin(w) + cos(p)*sin(eta)*cos(w)) +
            ez*(cos(eta)*cos(p)*cos(w) - cos(p)*sin(eta)*sin(w)) + l4*(cos(eta)*cos(p)
            *cos(w) - cos(p)*sin(eta)*sin(w));
239         d1 = yw;
240         R = sqrt(pow(xw,2)+pow(zw,2));
241         P = sqrt(pow(l3,2)+pow(l2,2));
242         alfa = atan2(zw,xw);
243         D = (-pow(P,2) + pow(R,2) + pow(l1,2))/(2*R*l1);
244         if (D>1) {beta=0;}
245         else {    beta = atan2(sqrt(1-pow(D,2)),D); }
246         t2= (alfa+beta);
247         z3=l1*sin(t2);
248         x3=l1*cos(t2);
249         gama = atan2(l3,l2);
250         delta = atan2(zw-z3,xw-x3);
251         t3 =(gama - delta);
252         R33 = sin(t3)*(cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p)) + sin(eta)*(
            cos(w)*sin(r) - cos(r)*sin(p)*sin(w))) - cos(t3)*(cos(eta)*cos(p)*cos(w) -
             cos(p)*sin(eta)*sin(w));
253         t4=-atan2(cos(eta)*(cos(r)*sin(w) - cos(w)*sin(p)*sin(r)) + sin(eta)*(cos(r)*
            cos(w) + sin(p)*sin(r)*sin(w)),(sin(t3)*(cos(eta)*cos(p)*cos(w) - cos(p)*
            sin(eta)*sin(w)) + cos(t3)*(cos(eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p)
            ) + sin(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w))));
254         t5=(atan2(sqrt(1- pow(R33,2)),R33));
255         t6=atan2(-(- sin(t3)*(cos(eta)*(cos(w)*sin(r) - cos(r)*sin(p)*sin(w)) - sin(
            eta)*(sin(r)*sin(w) + cos(r)*cos(w)*sin(p))) - cos(t3)*(cos(eta)*cos(p)*
            sin(w) + cos(p)*sin(eta)*cos(w))),  -(cos(t3)*sin(p)+cos(p)*cos(r)*sin(t3)
            ));
256         f2 = cálculo del índice de manipulabilidad
257       }
258       iter++;
259       if (f1>f2) {eopt=e1;
260         er=f1;}
261       else {eopt=e2;
262         er=f2;}
263       if (eopt !=0) {
264         ea =(1-sd)*100*((ef-ei)/eopt);
265         if (ea<0)
266         {ea=-ea;}  }
```

```
267  } while ( ea>es );
268
269  w=(w)*grados;
270  p=(p)*grados;
271  r=(r)*grados;
272  NSLog(@", %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f, %f,",x,y,z,w,p,r,eopt*grados,j
         ,er,d1,t2*grados,t3*grados,t4*grados,t5*grados,t6*grados);
273  NSString *stringJ1 = [NSString stringWithFormat:@"%4.3f",d1+1000];
274  NSString *stringJ2 = [NSString stringWithFormat:@"%4.3f",(t2*grados)+360];
275  NSString *stringJ3 = [NSString stringWithFormat:@"%4.3f",(t3*grados)+360];
276  NSString *stringJ4 = [NSString stringWithFormat:@"%4.3f",(t4*grados)+360];
277  NSString *stringJ5 = [NSString stringWithFormat:@"%4.3f",(t5*grados)+360];
278  NSString *stringJ6 = [NSString stringWithFormat:@"%4.3f",(t6*grados)+360];
279  NSString *msgEnviar = [NSString stringWithFormat:@"; %@; %@; %@; %@; %@; %@;!",stringJ1
         ,stringJ2,stringJ3,stringJ4,stringJ5,stringJ6];
280  label.text = msgEnviar;
281  socket = [[AbstractSocket alloc] init];
282  NSMutableData* respuesta1 = [[[NSMutableData alloc] init] autorelease];
283  @try {   [socket connectToHostName:@"148.224.56.98" port:59002];
284    isConnected = true;}
285  @catch (NSException *e) {
286    isConnected = NO;   }
287  if ([socket isConnected]) {
288    [socket writeString:msgEnviar];
289    while ( [socket readData:respuesta1] ) {}}
290  else {}
291  [socket close];
292 }
293 //
294 -(void)inclinacion {
295  switch (pos)
296  { case 1:{
297      label2.text=@"mandando posición p+";
298      p=p+1;}break;
299    case 2:{
300      label2.text=@"mandando posición p-";
301      p=p-1;}break;
302
303    case 3:{
304      label2.text=@"mandando posición r+";
305      r=r+1;}break;
306
307    case 4:{
308      label2.text=@"mandando posición r-";
309      r=r-1;}break;
310    default:
311    {}break;      }
```

```
312
313 // coloco al robot en una posición inicial
314 -(IBAction)inicio{
315   [datosTimer invalidate];
316   datosTimer = nil;
317   //movTimer = nil;
318   [timer invalidate];
319   timer = nil;
320   x= 500 ;
321   y= 1800 ;
322   z= 1230 ;
323   w= -50 ;
324   p= -85 ;
325   r= -20 ;}
326 // detiene el muestreo de datos
327 -(IBAction)stop{
328   [datosTimer invalidate];
329   datosTimer = nil;
330   [timer invalidate];
331   timer = nil;}
332 // Salida del programa
333 -(IBAction)salir{
334   exit(0);}
335 // Condiciones iniciales del programa
336 - (void)viewDidLoad {
337   l1=770;l2=100; l3=740; l4=100; ex=-223.2; ey=-3.4;ez=388.2;
338   xl=500;yl=1800;zl=1230;
339   acc =[UIAccelerometer sharedAccelerometer];
340   acc.delegate=self;
341   acc.updateInterval=0.1f;
342     [super viewDidLoad];}
343 @end
```

Bibliografía

[1] Carnegie Mellon University CMU. Project aura. http://www- 2.cs.cmu.edu/ aura/. Consultada el 20 de Agosto del 2010

[2] Hewlett Packard HP. Cooltown, http://www.cooltown.hp.com, Consultada el 20 de Agosto del 2010.

[3] Massachusetts Institute Technology MIT. Oxygen. http://oxygen.lcs.mit.edu. Consultada el 20 de Agosto del 2010.

[4] Pedro Neto, J. Norberto Pires, Accelerometer-Based Control of an Industrial Robotic Arm, The 18th IEEE International Symposium on Robot and Human Interactive Communication Toyama, Japan, Sept. 27-Oct. 2,2009.

[5] Oleg Alexandro, Cravioto García, José Jairo Ordaz Jiménez, Juan Manuel Ibarra Zannatha, Rafael Cisneros Limón, THuTouch: Sistema de Comunicación Humano-Humanoide Basado en un iPod Touch, XII Congreso Mexicano de Robótica (COMRob 2010).

[6] Zijing Lin, Max Q.-H Meng, Wanming Chen, Huawei Liang, Xin Liu, Design of a PDA-based Telerobotic System , Proceedings of the 2007 IEEE International Conference on Robotics and Biomimetics, December 15-18, 2007, Sanya, China.

[7] Carl Lundberg, Carl Barck-Holst, John Folkeson, Henrik I. Christensen, PDA interface for a field robot Proceedings of the 2003 IEEE/RSJ, Intl. Conference on Intelligent Robots and System, Las Vegas, Nevada, October 2003.

[8] Pierre Rouanet, Jérome Béchu, Pierre- Yves Oudeyer, A comparison of three interfaces using handheld devices to intuitively drive and show objects to a social robot; the impact of underlying metaphors, The 18th IEEE International Symposium on Robot and Human Interactive Communication Toyama, Japan, Sept. 27-Oct. 2,2009.

[9] Brad R Geltz, Jacob A. Berlier, James M. McCollum, Ph.D., Using the iPhone and iPod Touch for Remote Sensor Control and Data Acquisition, IEEE SoutheastCon 2010(SoutheastCon), Proceedings of the Publication Year 2010, Page(s): 9-12, IEE Conferences.

[10] Emmanuel Nuño Ortega, Luis Basáñez Villaluenga, Teleoperación: técnicas, aplicaciones, entorno sensorial y teleoperación inteligente, Universidad Politécnica de Cataluña, IOC-DT-P-2004-05 Abril 2004.

[11] Nilas Phongchai, A PDA-based Human-Robot Interaction using EMG-based Modified Morse code, Proceedings of the 2005 International Conference on Computational Intelligence for Modeling, Control and Automation, and International Confe Intelligent Agents, Web Tecnologies and Internet Commerce (2005) .

[12] Nilas Phongchai, A PDA-based Human-Robot Interaction for disabled persons using electromyography, IEEE CCECE/CCGEI, Ottawa,May 2006.

[13] Valero, A.; Randelli, G.; Botta, F.; Saracini, C.; Nardi, D.; Give me the control, I can see the robot!, Safety, Security and Rescue Robotics (SSRR), 2009 IEEE International Workshop.

[14] Tsuneo Yoshikawa, Manipulability of Robotic Mechanisms, The International Journal of Robotics Research, Vol. 4,No 2, Summer 1985.

[15] Tsuneo Yoshikawa, Translational and Rotational Manipulability of Robotic Manipulators, International Conference on Industrial Electronics, Control and Instrumentation series, IECON Proceedings, pp 1170-1175,Nov, 1991.

[16] Juan Manuel Ibarra Zannatha, Braulio Samuel Colmenero Mejía, Método de optimización global para el cálculo de la Cinemática inversa de robots redundantes, Departamento de Control Automático del CINVESTAV,2002.

[17] J. Kenneth Salisbury, John J. Craig, Articulated Hands: Force Control and Kinematic Issues. The International Journal of Robotics Research,1,pp 4-16,1982.

[18] C. Gosselin, J. Angeles. A global Performance Index for the Kinematic Optimization of Robotic Manipulators, Journal of Mechanical Design, 113, pp 220-226, 1991.

[19] Apple, iPhone Application Programming Guide , Manual, 2009.

[20] Dave Mark, Jeff Lamarche, Beginning Iphone Development, Exploring the Iphone SDK, apress,2009.

[21] Wei-Meng Lee, Beginning iPhone® SDK Programming with Objective-C, Wiley Publishing, Inc.,2010.

[22] González E., Notas sobre Teoría de Estimación Aplicada a la Robótica. CIEP-UASLP, 2010.

[23] Mark W. Spong, Seth Hutchinson, M. Vidyasagar, Robot Modeling and Control, JOHN WILEY and SONS, INC., first edition, 2005.

[24] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, Numerical Recipes in C, Cambridge University Press, second edition,1992.

[25] Chapra Steven, Canale Raymond, Métodos numéricos para ingenieros. Mc Graw Hill, cuarta edición, 2003.

[26] Fanuc, V7.30 R-30Ia Handling Tool Documentation MICROC73HYN04071E Rev A, 2007.

[27] A. M. Dudar,R.L. Whiterspoon, A High capability teleoperated vehicle for hazardous appplication, Third iasted international conference on robotics and manufactoring , Cancún, junio 1995.

[28] G.H Ballantyne, Robotic Surgery, telerobotic surgery, telepresence, and telementoring, Surgical Endoscopy, 16,1392-1402, Julio, 2002.

[29] Ji-Sun Kim, Denis Gracanin, Kresimir Matkovic, Francis Quek, iPhone/iPod Touch as Input Device for Navigation in Immersive Virtual Environments,IEEE Virtual Reality 2009, 14-18 March, Lafayette, Lousiana, USA.

[30] José María Sabater Navarro,Desarrollo de una interfaz kinestética paralela y experimentación en control de sistemas hápticos y teleoperados, Tesis de Maestría, Escuela Politécnica Superior de Elche, Junio, 2003.

[31] Dale A. Lawrence, Impedance Control Stability Properties in Common Implementations, IEEE International Conference on Robotics and Automation. Philadelphia, PA, 2, pp. 1185-1190, 1988.

[32] Neville Hogan. Impedance Control: An Approach to Manipulation: part I-Theory. Department of Mechanical Engineering and Laboratory for Manufacturing and Productivity, Massachusetts Institute of Technology, Cambridge, Mass. March,vol. 107, 1-7, 1985.