



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**TELEOPERACIÓN DE UN MANIPULADOR
DE 5 GRADOS DE LIBERTAD
UTILIZANDO PLATAFORMA VIRTUAL**

T E S I S

Que para obtener el título de

INGENIERO MECATRÓNICO

P R E S E N T A

DANIEL ALBERTO ORTIZ PÁMANES

DIRECTOR DE TESIS
M.I. OCTAVIO DÍAZ HERNÁNDEZ



CUIDAD UNIVERSITARIA,
MEXICO, D. F.

AGOSTO 2014



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Resumen

Se discute en detalle el análisis y solución de la cinemática inversa de un manipulador en configuración serial de seis grados de libertad en espacio restringido, algoritmo que se implementó y probó en un robot prototipo en configuración de manipulador serial de cinco grados de libertad, con resultados exitosos. Se presentan las herramientas teóricas de análisis matemático necesarias para el análisis cinemático del robot. El proyecto consistió en la implementación y puesta en marcha de una plataforma de realidad virtual para teleoperar el robot prototipo.

Las funciones de la plataforma consisten en establecer, monitorear y supervisar la comunicación con el robot manipulador y con otros dispositivos de captura de posición y orientación espacial, como un mecanismo maestro instrumentado, acelerómetros, o unidades inerciales, también en resolver en tiempo real la cinemática inversa de las articulaciones, aplicar restricciones cinemáticas virtuales; además de diversos algoritmos de navegación y control cinemático. Se discute la importancia de la plataforma para la teleoperación de un sistema mecatrónico de acuerdo a trabajos relacionados.

Abstract

The present work deals with the inverse kinematic analysis of a serial manipulator with six degrees of freedom, and the search of solutions in the constrained workspace. The implementation of the inverse kinematic problem was tested on a robotic prototype, in a naive obstacle evasion experiment with successful results. The implementation consisted of three parts: the development of a user interface with a virtual environment for supervision, command, control and communication with the robot; the development of a serial communications protocol, and the development of the program hosted in the microcontroller with direct access to the prototype.

Contenido

1	Introducción	1
1.1	Historia	1
1.2	Línea de tiempo	1
1.3	Definición de teleoperación	2
1.4	Espectro de control	4
1.5	Control directo	5
1.6	Control supervisado: computadora	6
1.7	Control supervisado: incertidumbre	6
1.8	Control supervisado: telerobot	7
1.9	Sensores situacionales	9
1.10	Actualización interactiva del modelo	9
1.11	Definición de SA	10
1.12	Incertidumbre	11
1.13	Carga de trabajo, formalización	12
1.14	Telepresencia	12
1.15	Interfaces	13
1.16	Diferencias entre operación y teleoperación	14
1.17	Diferencias entre teleoperación y control supervisado	14
1.18	Modelo de un sistema teleoperado.	15
1.19	Aplicaciones en teleoperación.	16
1.20	Descripción del proyecto.	16
1.21	Metas	17
1.22	Objetivos	17
1.23	Descripción del experimento.	17
2	Análisis cinemático del manipulador	18
2.1	Representaciones de orientación	18
2.2	Matriz de rotación	18
2.3	Ángulos de Euler.	20
2.4	Determinación de la orientación Tait-Bryan de un sistema A relativo a B.	22
2.5	Modelo cinemático por Denavit-Hartenberg.	26
2.6	Solución geométrica cinemática inversa de 6GDL.	36
2.7	Cinemática inversa en espacio de trabajo restringido.	39
2.8	Control cinemático	43
2.9	Acoplamiento virtual de objetos.	43
3	Plataforma virtual	45
4	Experimentación y resultados	48
5	Conclusiones	53
6	Apéndices	56
6.1	Deducción analítica de cinemática directa	56
6.2	Firmware del protocolo en Arduino	61
7	Referencias	70

Agradecimientos

A mi familia, a mis amigos, y a mis profesores.

Los autores agradecen el apoyo brindado por la DGAPA, UNAM, a través del proyecto PAPIIT IN117614, con título: "ROBÓTICA INTUITIVA, ADAPTABLE, REACTIVA, HÍBRIDA Y MÓVIL APLICADA AL SERVICIO, EL RESCATE Y LA MEDICINA" durante la realización de este trabajo.

Introducción

1.1 Historia

El origen de la palabra autómatas proviene del latín, y éste del adjetivo griego “*αυτοματος*” (automatos), “que se mueve por sí mismo”, “que obra espontáneamente”. Automatos contiene la raíz “auto” que significa “uno mismo”, “por sí solo”, “ego” y la raíz “matos” que significa “movimiento” o “voluntad”

La automatización históricamente consistió en utilización de autómatas en actividades productivas o industriales, en vez de entretenimiento o investigación. Esto sucedió en la segunda mitad del siglo XVIII en Francia, con la invención de telares mecánicos, prosiguiendo su asimilación industrial a partir de entonces.

Teleoperación y control remoto iniciaron como la misma forma de control. El primer dispositivo controlado remotamente por radio fue demostrado en 1898 por Nikola Tesla en la Primera Exhibición Eléctrica en el recién reinaugurado Madison Square Garden en Estados Unidos. Consistía en un navío miniatura denominado por su autor como “*Teleautomaton*” [35].

El término **robot** empezó a utilizarse a partir de 1920 después de que fuera popularizado por la obra de teatro de Karel Capek y por el escritor de ciencia ficción Isaac Asimov.

Los robots industriales evolucionaron a partir de los teleoperadores, como una forma automática de éstos controlada por computadora [33](Spong, p.3).

Se destacan a continuación antecedentes históricos, sintetizados de [18][Kurfess 1.1], [33](Spong, pp. 3-6), [4](Barrientos, pp. 3-6), [16](Goodrich pp.204-210) que llevaron a la invención y utilización de teleoperadores y robots. Se entrelazan historia de control, historia de computación, automatización y telecomunicaciones.

1.2 Línea de tiempo

Año / Periodo	Evento
100 a.C.	Mecanismo de Antikythera. Reloj astronómico. Se le atribuye origen griego, descubierto en 1901.
85	Mecanismos animados de Herón de Alejandría , movidos o actuados a través de dispositivos hidráulicos, poleas, y palancas.
700 - 1500	Autómatas de la cultura árabe.
1204 - 1282	Hombre de Hierro de Alberto Magno.
1241 - 1294	Cabeza parlante de Roger Bacon.
1352	Gallo de Estrasburgo. Parte de un reloj de la torre de la catedral de Estrasburgo.
1452 - 1519	León mecánico de Leonardo da Vinci.
1600 – 1700	Ingenios mecánicos creados por artesanos del gremio de la relojería para entretenimiento.
1745	Construcción del primer telar mecánico completamente automático, por Jacques Vaucanson (1709 - 1782), autor de varios autómatas.
1700	Hargreaves, hiladora giratoria.
1779	Crompton, hiladora mecánica.
1785	Cartwright, telar mecánico.
1801	Telar de Joseph Marie Jacquard. Utilizaba una cinta de papel perforada como programa para las acciones de la máquina.
1898	Tesla, Teleautomaton.

Año / Periodo	Evento
1921	Karel Capek origen de la palabra robot proveniente de la palabra eslava "robota", estreno de la obra checa Rossum's Universal Robots en el teatro nacional de Praga.
1940	Raymond C. Goertz, teleoperador mecánicamente acoplado para manipular material radioactivo.
1943	Asimov, escritor de ciencia ficción. Leyes de la robótica.
1954	R.C. Goertz. Teleoperador electromecánico. Incorporación de servocontrol y tecnología electrónica. Comunicación alámbrica Robots móviles, Robots industriales.
1954	George Devol, creación de la empresa Universal Automation.
1956	Devol y Engelberger. Primer compañía fabricante de robots: Unimation. Primer robot industrial instalado en una línea de producción por General Motors en una planta ubicada en Trenton Nueva Jersey.
1962	Utilizado en soldadura por puntos y extracción de piezas de fundición por inyección [4](Barrientos p.6) [33](Spong, p.6).
1963	Desarrollo del primer sistema de visión robótica.
1969	Yaskawa, definición de mecatrónica: la integración sinérgica de mecánica y electrónica.
1971	Desarrollo del brazo de Stanford en la Universidad de Standford por Victor Scheinman
1971	Formación de la asociación de robots industriales en Japón: JIRA (Japan Industrial Robot Association).
1974	Kawasaki compra el diseño del robot Unimation.
1974	La compañía Cincinatti Milacron introduce al mercado el robot T ³ con control por computadora.
1978	La compañía Unimation introduce el robot PUMA.
1979	El diseño de robot tipo SCARA es introducido en Japón.
1983	Fundación de Adept Technology.
1987	Fundación de la asociación de robótica y automatización de IEEE (Institute of Electrical and Electronics Engineers).
1988	Primer conferencia internacional en robots inteligentes y sistemas (IROS) en Japón.

1.3 Definición de teleoperación

El término **teleoperación** está formado por dos palabras: El prefijo griego "**tele**", que significa remoto y el sufijo latino "**operatio**" que significa algo hecho. [6](Basáñez Suárez, p.450); La definición de teleoperación según [11](Cui p. 1) es:

"Teleoperation are **means** to operate a robot using human intelligence, which requires the availability of adequate **human-machine interface**".

Entonces **teleoperación** son los **medios** y **métodos** para **operar** (controlar) un robot utilizando e integrando las funciones de la inteligencia o supervisión humana, lo que **requiere** de la disponibilidad de una adecuada **interfaz** humano-máquina (**HMI**) ya sea para **extender** las capacidades sensoriales o manipulativas del operador humano a distintas escalas de fuerza y precisión, o para **automatizar** y **supervisar** un proceso.

Estos medios y métodos lo constituyen el equipo y soporte tecnológico, como son las interfaces; los paneles de control, los paneles de instrumentos o dashboard, que pueden ser digitales o virtuales; los sistemas de información, realidad virtual o realidad aumentada, los protocolos de comunicación de control y datos, los canales o enlaces o ciclos de control y datos, y los sensores, los instrumentos, servomecanismos, entre otros

Teleoperación es semánticamente idéntico de control remoto. Se refiere a la interacción remota con una máquina en un entorno no estructurado para realizar o ejecutar alguna acción, tarea, función u operación desde cierta distancia, o **comunicar** los comandos de control de dicha acción o tarea, y esperar la confirmación de su ejecución.

Teleoperación es entonces una forma de **telecomunicación** en que existe principalmente un flujo de información de control, con origen en la estación de control y como destino el teleoperador. El flujo de información proveniente del teleoperador hacia la estación de control, cuando está presente, es un enlace de datos. En la implementación, los canales y **protocolos** de comunicación para cada enlace pueden ser distintos.

Un **teleoperador**, según Sheridan, es cualquier herramienta, máquina o dispositivo que permite a un operador humano, **explorar** un entorno remoto, y manipular e **interactuar** con objetos, o modificar y controlar aspectos del entorno [28]. La interacción con objetos es mecánica, extendiendo las capacidades motrices y biomecánicas de una persona.

Esta definición coincide con la de [11](Cui, p. 1):

“Teleoperator is a machine that enables a human operator to move about, **sense** and mechanically manipulate objects at a distance...”

Entonces, **el teleoperador** constituye el **agente remoto** a través del cual un operador recibe información sensorial del entorno, reacciona e interactúa ante estímulos y ejecuta **tareas**.

Ésta manipulación obedece a tareas como ensamble, inspección, mantenimiento, reconocimiento, vigilancia, etc.

De estas definiciones se distinguen las funciones de **explorar, percibir** y recibir información **sensorial** del entorno. Estas funciones son formas de comunicación. Implícito en ellas, está la **confirmación** del evento de manipulación, es decir, realimentación: el cierre del ciclo de control. Una vez que se piensa en términos de realimentación, se puede dividir ésta en tipos: visual, acústica, táctica (“haptic”). La realimentación además es continua, y la información recibida o extraída se integrará en un mapa o modelo del entorno, dando lugar a **SA**, discutido más adelante.

La implementación de realimentación hace siempre necesaria la existencia de un canal de comunicación de cualquier tipo para para la determinación o medición del **estado operacional**, es decir, la **actualización** continua e interactiva del mapa del entorno, posiblemente implementado en un modelo (programa) computacional.

La función principal de un sistema de **teleoperación** es **asistir** al operador para realizar tareas de alto nivel, complejas, e inciertas en ambientes peligrosos, **no estructurados**, remotos, de difícil acceso, **desconocidos**, a distinta escala.

Aunque no es explícito en las definiciones, si el conocimiento sobre el entorno es incierto, la primera tarea sería reducir esa **incertidumbre**: obtener un modelo o mapa del entorno, integrando información operacional y situacional. Y luego visualizar el modelo o mapa.

Teleoperación está asociado a robótica y a robots móviles, porque estas aplicaciones cumplen con las condiciones de incertidumbre y movilidad

En las aplicaciones, alcanzar autonomía completa es de considerable dificultad debido a la incertidumbre. Por eso se requiere de supervisión humana, por lo que tiene que **integrarse** un humano al ciclo de control. Esta integración se logra mediante la inclusión de una interfaz de usuario.

Las ventajas de un sistema **teleoperado** entonces, son el aprovechar eficientemente las distintas capacidades del operador y la máquina y sus componentes, y se manifiesta en la separación y administración de tareas: como inspección, navegación, acumulación de información, etc.

El operador humano se encarga de tareas de alto nivel, como toma de decisiones estratégicas, reprogramación remota del sistema operativo del teleoperador, actualización de código, cambios en la misión, clasificación de eventos y objetivos, etc., delegando decisiones de bajo nivel y altamente modeladas como control directo, al módulo de inteligencia artificial del teleoperador

1.4 Espectro de control

Teleoperación es considerado por algunos autores como idéntico de **control remoto**; el polo opuesto de **control automático**. Otros como [13](Fong et al., p. 9), consideran **teleoperación** como todo un **espectro** de distintos **modos** o formas de control incluyendo control automático.

Históricamente, **teleoperación** y **control remoto** iniciaron como la misma forma de control, y sólo conforme nuevos sistemas computacionales más capaces fueron introduciéndose en el mercado e incorporándose en el sistema teleoperado, teleoperación fue diferenciándose en un espectro de aplicaciones con diferente nivel de automatización.

El **espectro de control** comprende todas las formas de control empezando con control **directo**, pasando por control **supervisado**, hasta control **automático**. Estos distintos **modos de control** se diferencian uno de otro en cuanto al nivel de automatización; resultado de la integración con tecnologías de información y el cierre de ciclos de control por la computadora, y al nivel de interacción e integración con un humano, es decir, los modos de control también están altamente correlacionados con los tipos y capacidades interactivas y computacionales de las interfaces, como se reitera en [28](Sheridan_92, pp. 1, 4, 9, 65), [13](Fong, pp. 9, 13-15) y [16](Goodrich, p. 216).

Sheridan separa el espectro de control en tres modos: control **directo**, **supervisado**, y **automático** [28](Sheridan_92, pp. 2, 65):

Control directo o también control **manual** (considerado también como **control remoto**), es el modo menos sofisticado dentro del **espectro de control**; constituye una forma de control que se considera desprovista de un modelo formal. De otra forma este modelo se implementaría en una computadora y se añadiría una interfaz para interactuar con el sistema.

En **control directo**, la operación está excluida de cualquier forma de **automatización**, puesto que no se cuenta con una **interfaz multimodal** capaz de **telepresencia** y **supervisión**, ni con la integración e interacción con un **sistema computacional** intermediario y la **asistencia** provista por éste, debido en parte, a que los modelos tanto del dispositivo como del entorno son **informales**, es decir, el **modelo operacional** sobre el cual se toman decisiones de control, permanece exclusivamente en la mente del operador y no hay forma de comunicarlo (o delegarlo) a un posible sistema computacional **durante** la operación. Sobre el operador recae todo el control y la coordinación de los grados de libertad del dispositivo, y también la carga de tareas por ejecutar.

La asistencia computacional consiste en **telepresencia** y visualización, monitoreo, registro de telemetría, mapeo interactivo del entorno, realidad aumentada o ambientes virtuales, generación de trayectorias, simulación, ejecución semiautomática, control restringido, etc. [28](Sheridan_92 p. 65).

Cuando en **control directo** se incluye un sistema computacional, el control se realiza **a través** de una computadora; las funciones provistas por ésta constituyen solamente en la forma más básica de **telepresencia**: la administración de los canales de realimentación visual y auditiva. Es decir, ninguna tarea puede ser delegada a la computadora y operarse automáticamente.

Una vez que se ha integrado un sistema computacional, el control puede aumentarse si se provee mayor asistencia computacional: como la inclusión de una interfaz especializada, con telepresencia, entornos virtuales, etc. Cuando estas capacidades computacionales e interactivas están presentes, control **directo** se vuelve control **supervisado**, debido en parte a que algunas tareas pueden delegarse, es decir ejecutarse automáticamente, dado que se han estudiado, formalizado, modelado e implementado en el sistema computacional.

La distinción y evolución de **control supervisado** a **control automático** es gradual y depende en gran medida de la formalización de los **modelos operacionales** y su implementación en **modelos computacionales**, y también de la **actualización interactiva** de dichos modelos: ya sea por la comunicación humana a la **computadora** a través de la **interfaz**, de tareas y objetivos identificados y seleccionados por un operador humano, o reconocimiento automático de tareas y objetivos por medio de **inteligencia artificial**. La interacción en **control supervisado**, es ahora con información, además de la interacción de **control directo** con la máquina remota. Es por esto que **control supervisado** también se denomina **multimodal**, dado que permite tanto **control directo** como control **semi-automático**.

Cuando el sistema es semi-automático, algún agente es responsable de la operación, supervisión, control y estabilización. De aquí se pueden deducir muchos requerimientos funcionales de la interacción: el sistema debe comunicar su estado a un operador humano; y el humano debe comunicar tareas y objetivos al sistema. Estas funciones podrían enumerarse como **telepresencia**; **telemetría** y **supervisión**; comunicación y monitoreo, requerimientos que son satisfechos por la inclusión de una interfaz y un sistema computacional. En este caso, la comunicación es entre operador y computadora, facilitada por una interfaz y se refiere a la especificación **interactiva** de objetivos, tareas y restricciones. Para que la especificación de tareas sea posible, es necesaria la implementación previa de un **modelo computacional**.

1.5 Control directo

Otro conjunto de características que distinguen al **control directo** de otras formas más automáticas de control es que no es necesaria la existencia de canales dedicados de **realimentación**, ni la existencia del soporte tecnológico en el **teleoperador** para estas funciones. Como no existe realimentación, o no se cierra el ciclo de control; el tipo de control es abierto, además de informal. Pues en cualquier forma de automatización, se deben cerrar los ciclos de control

Los comandos de control se transmiten sobre un canal de comunicación independiente y unidireccional.

En este caso, el dispositivo bajo control se encuentra bajo el alcance visual no instrumentado de un operador; sin asistencia de instrumentos ópticos. Por lo tanto, la capacidad de decisión del operador es limitada; dado que su **conocimiento del entorno** es limitado.

Sin canales de realimentación, no hay forma de determinar la **situación operacional** del dispositivo bajo control, es decir, no se cuenta con información precisa acerca de los **grados de libertad controlables**: no se miden, registran, confirman o verifican las variables y operaciones realizadas por el dispositivo.

El canal de información sobre el cual se codifica y transmite la medición calibrada de variables o parámetros **internos** del teleoperador, se denomina **telemetría** y el soporte tecnológico se denomina **instrumentación**. Las mediciones provienen del **teleoperador** y se comunican hacia la estación de control.

Dado que no es necesario que exista este canal de comunicación, no es posible determinar con precisión la posición, ni orientación; ni velocidad del dispositivo; parámetros que se utilizan en tareas como navegación; ni registrar el desempeño

El operador además no cuenta con realimentación de alto nivel, como video o audio, para aumentar su **conocimiento del entorno**; no cuenta con el **punto de vista** del dispositivo bajo control, por ejemplo, no conoce los obstáculos que están cerca, sino que deduce esta información (posición, orientación, velocidad del dispositivo) a partir de lo que **alcanza a ver**, usando como **referencia** el paisaje o entorno. Reduciendo el rango de movilidad de teleoperación, pues se requiere la **presencia** del operador para que **tome cuenta** de obstáculos y otros elementos ambientales. La **telepresencia** entonces es una forma remota de presencia, y el conjunto de funciones que la **telepresencia** debe proporcionar al operador son la captura de información visual y acústica, **acerca del entorno** y no la captura de información de las variables internas del dispositivo remoto (**telemetría**). La **telepresencia** es la base cognitiva sobre la que el operador construye, actualiza y registra mentalmente un mapa o modelo del entorno. Cuando este modelo se encuentra implementado en un sistema computacional, la **telepresencia** toma forma de realidad aumentada o ambientes virtuales.

Sin este importante canal de realimentación entonces, el operador no puede tomar decisiones de alto nivel que le ayuden a cumplir tareas de manipulación o navegación o inspección remotas, pues carece de la

información subyacente necesaria. En **control directo** tampoco hay formalización o actualización de estado en un **modelo computacional**

Por todo lo anterior, el rango y robustez de operación que distinguen al **control directo**, es decir, qué tan estable, preciso y repetible es el control, es considerablemente menor, nulo, de control abierto.

1.6 Control supervisado: computadora

La discusión anterior identifica requerimientos operativos en teleoperación, como la realimentación de **información situacional**, que son resueltos con la integración de soporte tecnológico: un sistema de captura de información del ambiente, un enlace de telecomunicaciones bidireccional, una **interfaz de usuario** con **telepresencia**, y un **sistema computacional** que coordinará todos los sistemas anteriores

En **control supervisado** la inclusión de los sistemas anteriores extiende la movilidad y capacidades operativas del dispositivo bajo control permitiendo ser operado fuera del alcance visual del usuario, volviendo posible salvar barreras físicas en el entorno, distancia y tiempo, siempre y cuando exista un enlace robusto de comunicación [13].

El dispositivo remoto utiliza medios electrónicos para establecer un **enlace de telecomunicación** para recibir instrucciones de control y transmitir información codificada acerca de su situación espacial, operacional y situacional a un operador, información que es procesada, integrada y después desplegada en una terminal.

El trabajo de integración de información en un **mapa o modelo** y toma de decisiones se delega al operador humano que toma las decisiones por parte del robot, utilizando la información situacional comunicada por éste.

Las tareas altamente modeladas, y por lo tanto automatizables, se identifican por parte de un operador y se comunican y delegan al sistema, es decir, se actualizan en un mapa o modelo o programa. Su ejecución es supervisada y monitoreada. Entonces, tareas de alto nivel (objetivos, trayectorias) se separan de las de bajo nivel (coordinación y estabilización), lo que induce la división y administración de carga de trabajo entre operador y computadora.

El **estado operacional** es el estado de las variables internas del teleoperador, incluyendo las variables controlables.

La **información situacional** incluye variables externas ambientales, como video, audio, temperatura, fuerzas, contacto, velocidad y presión del viento, etc. Esta información es integrada para proveer al operador de **telepresencia** y que éste aumente **su conocimiento del entorno**, y tome en cuenta las circunstancias ambientales durante tareas de alto nivel como navegación (determinación de postura y generación de trayectorias), identificación y selección de objetivos, etc.

1.7 Control supervisado, Incertidumbre

En **control supervisado**, a pesar de contar con la integración de un **sistema computacional** y una **interfaz**, aún se considera que el sistema teleoperado no es lo suficientemente autónomo; es decir, el sistema por sí mismo no es capaz de generar y actualizar **sus propios modelos del entorno**, por lo que se vuelve necesaria la existencia o presencia de un operador humano en una terminal de control. Sobre él recae la tarea de mantener y actualizar los **modelos y mapas del entorno** y administrar la **carga de trabajo**. El operador es el responsable de ejecutar y supervisar el control, quien puede tomar desde decisiones de alto nivel como navegación, hasta control directo. En modo directo, el operador es responsable de la estabilidad y coordinación de todos los grados de libertad del sistema teleoperado.

Las tareas de alto nivel son traducidas en componentes de bajo nivel y delegados al sistema. Estos componentes de bajo nivel son tareas modeladas e implementadas en un programa computacional. El operador realiza la traducción de tareas en su mente, utilizando un **"programa mental"** adquirido por experiencia o entrenamiento. Estas tareas corresponden con **objetivos identificados** en el sitio remoto, y se sintetizan en control directo del teleoperador.

El operador puede, si lo desea, controlar cierto subconjunto de grados de libertad, dejando que la computadora controle y coordine automáticamente los demás, de acuerdo a objetivos de alto nivel. Esto distingue a **control supervisado**: cuando existe una implementación computacional del mapa del entorno y del modelo operacional, y cuando este mapa es actualizado por un humano.

Otras tareas características de control supervisado son entonces la implementación de modelos virtuales; la formalización de modelos empíricos, es decir, la **actualización interactiva** de modelos implementados, o la implementación o programación de nuevos **modelos computacionales** o extensiones a los anteriores. Esto sucede debido a que el **modelo operacional** está en constante crecimiento: empieza siendo informal, conforme se estudia se vuelve parcialmente formal, y durante la formalización, también puede estar parcialmente implementado.

Las tareas relacionadas con la actualización o implementación del modelo operacional se consideran de alto nivel. La introducción de **Inteligencia artificial** en el sistema, se realiza para automatizar estas funciones, es decir, reconocimiento de objetos, actualización del mapa de entorno, etc., dando lugar a los distintos tipos de control supervisado como se discute en [28](Sheridan_92 p.65).

Una de las tareas iniciales del sistema **teleoperado** es obtener un **modelo** o **mapa del entorno**; reducir la **incertidumbre**, por lo que el telerobot debe contar con tecnología de soporte para instrumentación, y componentes y sensores ambientales o situacionales dedicados, también denominados **sensores multimodales**. Del lado maestro, se debe contar con interfaces, tecnologías de información, etc.

Sensores multimodales son aquéllos que sirven o son utilizados en distintos modos o niveles de control, es decir, un sensor multimodal sirve para tareas de alto nivel, como actualización del **modelo computacional** del entorno; su función es la de capturar y proveer **información situacional** del entorno; proveer de **telepresencia**, a diferencia de los sensores que son parte de la instrumentación y telemetría que se utilizan en funciones de control de bajo nivel. Es precisamente por la dificultad del procesamiento y clasificación de información situacional, como video o audio, que se requiere la supervisión de un operador humano

En **control supervisado**, la **reducción de incertidumbre** se realiza en dos pasos: primero, la asimilación de **telepresencia** por parte del operador, y posteriormente la actualización del **modelo computacional**, que consiste en la comunicación o especificación de los objetivos identificados en el **mapa** o programa.

Inicialmente el programa computacional tiene absoluta incertidumbre. El humano reduce su propia incertidumbre a partir de la **telepresencia**, y formaliza sus hallazgos, registrándolos o actualizándolos en un mapa o modelo o programa computacional, es decir, el humano reduce la incertidumbre del sistema computacional una vez que ha reducido su propia incertidumbre. Descarga, Alivia, Delega su carga de trabajo como describe [28](Sheridan_92 p.65)

En resumen, en control supervisado se requiere la presencia continua de un operador. El operador interactúa con información, supervisando tareas especificadas a la computadora e identificando objetivos a partir de telepresencia que son comunicados o actualizados en un mapa computacional.

1.8 Control supervisado: telerobot

Un **telerobot** es una subclase de teleoperador altamente automatizado. En un sistema automático, las decisiones operativas quedan a cargo de un módulo local de control de inteligencia artificial; un dispositivo electrónico (microcontrolador o sistema embebido) **previamente programado** (en el caso de **un telerobot**). Al estar previamente programado, significa que se implementó un **modelo** en forma de programa que será utilizado durante la operación. Este modelo es estático y no se actualiza automáticamente. La actualización por reprogramación queda a cargo de operadores humanos. Funciones facilitadas por una **interfaz de supervisión**, en la estación de control.

Un **telerobot**, entonces, es un robot que recibe instrucciones, comandos de control, objetivos o actualizaciones de estado, desde una estación remota; instrucciones provenientes de un operador humano a través de un sistema computacional y que ejecuta acciones en tiempo real sobre el entorno a través de

servomecanismos y actuadores, **independientemente**, valiéndose de información obtenida a partir de sensores y procesada por subsistemas de control.

Estas acciones son planificadas y supervisadas, por lo que el sistema debe contar con una interfaz multimodal en la estación de control.

El **telerobot** por lo tanto, también posee tecnología de soporte para establecer un **enlace de telecomunicación**. Por ejemplo, robots exploradores espaciales. En este caso, las distancias a cubrir por el enlace de telecomunicación son muy grandes, y las ventanas de comunicación pequeñas, esto es, el enlace no es estable. Además, el retraso de comunicación es tan largo que no permite una teleoperación directa o en tiempo real, provocando inestabilidades [29].

En aplicaciones espaciales, se refleja uno de los mayores problemas en **teleoperación**, que es el retraso en la comunicación debido a distancias en la transmisión y límites en el procesamiento de información, y disponibilidad de ancho de banda. Esto motivó el incremento de autonomía del **teleoperador** para volverlo **telerobot**. Se observa la vital relación entre **teleoperación y telecomunicaciones**

En un **telerobot**, las condiciones de operación son distintas a otros sistemas automáticos: se sigue considerando que no se posee información del entorno remoto, además se parte del requerimiento de movilidad e interacción con el entorno. Lo que hace necesario el uso de **sensores situacionales** y una **interfaz con telepresencia** para que se reduzca dicha incertidumbre.

Ya que la selección de objetivos **depende del entorno**, del **SA**, y esta selección está a cargo de agentes inteligentes, la información del entorno remoto debe comunicarse e integrarse en un mapa o **modelo** virtual, y después seleccionar objetivos y metas.

En la estación de control, el usuario humano tiene el rol de supervisor, es decir, monitorea, recibe e integra información acerca del estado operacional, para aprobar o descartar las distintas decisiones o acciones del módulo de control del telerobot.

Éste módulo de control local, utiliza de forma limitada **información del entorno**, para planear acciones e identificar y reportar riesgos en la misión. Esta información puede ser **comunicada** a módulos superiores locales, es decir, procesada, o reportada a la estación de control. Los módulos superiores locales podrían estar provistos de capacidad computacional suficiente para hospedar programas de inteligencia artificial, e integrarían información del entorno en un **modelo computacional** de forma automática, sin la intervención inmediata de un operador humano (o con moderada asistencia), dado que el enlace de telecomunicación con la estación de control puede ser intermitente,

El **telerobot** entonces, también debe poseer una representación o mapa virtual del entorno, que debe ser actualizado automáticamente, por lo que además de inteligencia artificial debe contar también con **sensores situacionales**

Las tareas de alto nivel (planeación, actualización en un modelo) también pueden delegarse a la estación de control, para ello en la estación de control se utilizan los reportes recibidos del **telerobot**. Estas tareas pueden estar **compartidas** entre el operador humano y la computadora dependiendo su nivel de formalización, y las capacidades ergonómicas e interactivas de la interfaz. La delegación de tareas de alto nivel puede deberse a que existen menos limitaciones computacionales en la estación de control.

En **control supervisado**, la interfaz con el sistema computacional y la interfaz con el teleoperador se fusionan en una sola.

El control **automático** puede ser en distintos niveles dentro del mismo sistema teleoperado: por ejemplo, nivel servocontrol. Para que el control pudiese ser de alto nivel y coordinado; se necesitaría de **inteligencia artificial** e incluiría **visión por computadora** y de **sensores situacionales**.

Las ventajas de un modelo interactivo es que éste se adapta rápidamente ante eventos en el entorno, y el sistema responde con la misma velocidad.

1.9 Sensores situacionales

El propósito de un sistema **teleoperado** es controlar, influir o manipular en cierta medida algunas de las variables y objetos del entorno, es decir variables situacionales, como los variables de postura de un objeto. Dependiendo de las capacidades de la interfaz y el teleoperador, estas variables pueden ser manipuladas en un rango operativo. Por ejemplo, posición, temperatura, estructura mecánica, etc.

Se puede diferenciar entre una aplicación de automatización industrial como un robot industrial o control de proceso, y un **telerobot**, en que las aplicaciones industriales no requieren de sensores dedicados a proveer información del entorno, pues se encuentran por diseño en entornos controlados, mientras que en un **telerobot** estos sensores son indispensables. Además, los dispositivos del sistema industrial se encuentran altamente modelados, pudiendo ser supervisados y controlados automáticamente.

Este requerimiento responde a la **incertidumbre** sobre el entorno, y se manifiesta sobre el **telerobot** en medios y métodos que capturen, almacenen, procesen y comuniquen **información situacional** del entorno, y métodos que interactúen y administren esta información: como protocolos, software y enlaces de datos, entre otros.

Estos sensores podrían denominarse situacionales, dedicados a medir variables del entorno, es decir variables no controladas, pero que proveen de conocimiento del entorno.

Su función es capturar y comunicar información situacional del entorno a la estación de control, con el propósito de que un operador identifique y clasifique objetos, seleccione tareas y establezca objetivos. Los objetivos y metas constituyen la información operacional y son siempre relativos al entorno, objetivos cuyos parámetros son a su vez situacionales y deben ser monitoreados constantemente, esto es, actualizados en un mapa.

1.10 Actualización interactiva del modelo

La selección y clasificación de objetivos son tareas de alto nivel, presentes en todo el espectro de control, y que en control supervisado y telerobótica se pueden delegar a la computadora siendo posible su actualización en un mapa computacional del entorno, para ser considerados de forma automática durante la operación.

La captura, transmisión y procesamiento de información visual, acústica, táctil (“haptic”), térmica, cinemática, en conjunto operacional, y la integración de esta información operacional en un **modelo virtual interactivo**, y la frecuencia con que esto suceda incrementa las capacidades tácticas del sistema para desempeñarse ante las tareas impuestas (volverse automático), puesto que el **modelo** del entorno remoto se incrementa.

Toda esta **información situacional** puede estar sujeta a procesamiento, formalización y modelado extensivo: reconocimiento, clasificación, organización, integración; por parte de un operador humano, o de forma automática por inteligencia artificial, basándose en información histórica (registro histórico por monitoreo) y en bancos de datos.

En un telerobot, las decisiones de control, y también las decisiones de alto nivel o nivel tarea, se basan en la determinación del estado operacional o situacional actual a partir del cual se fluye hacia un estado operacional deseado. Todo estado deseado es relativo al entorno; y para esto, se necesita partir de la información del entorno. Durante la transición de estados, se toma en cuenta el **modelo** o mapa, incompleto y aproximado, del entorno y las capacidades dinámicas (capacidades mecánicas diseñadas e implementadas: actuadores, mecanismos, elementos terminales) del teleoperador o **telerobot**, junto con las capacidades de la **interfaz** y el sistema computacional en la estación de control.

El **modelo del entorno** entonces, tiene **importancia operativa**, debe ser **interactivo**, para que pueda ser usado durante la operación. La interactividad del modelo se manifiesta inicialmente como informalidad, es decir, el modelo crece y existe dentro de la mente del operador (en forma de carga de trabajo). Si el modelo es informal, entonces el sistema requiere de una fuerte interacción, integración y comunicación con el humano, comunicación que es facilitada por una **interfaz** con elevada **telepresencia**, de modo que éste construya, mantenga, actualice y utilice su modelo informal de la forma más eficiente posible (lo comunique, o a partir de él, genere comandos de bajo nivel), sin perder la concentración, ya que dentro de su mente se encuentra el **registro del estado situacional** y también todas las operaciones a realizar.

Esta interacción, entre humano e interfaz, debe ser intuitiva; sin consumir o incrementar la **carga de trabajo**, o mantenerla estable y al mínimo, ya que el modelo mental es frágil, pero de suma importancia operacional.

El operador entonces está encargado de la determinación del estado operacional y el registro del estado situacional a partir de la **telepresencia**. Debe integrar información recibida del entorno remoto en su modelo mental. La interacción debe ser intuitiva (sin consumir o aumentar la **carga de trabajo**, o de carga de trabajo mínima). Sólo si una parte del modelo operacional ese encuentra implementada en un sistema computacional, la carga de trabajo podrá ser dividida entre el humano y computadora; dejando decisiones, tareas, funciones y modelos informales pero de alto nivel a cargo del operador humano. Dependiendo de la formalidad del **modelo**, esto es, su representación computacional, el sistema podrá cerrar sus propios ciclos de control incrementando su nivel de automatización.

La representación computacional toma la forma de entorno virtual, y se justifica por la facilidad de interacción con el entorno remoto una vez modelado.

1.11 Definición de SA

El estado operacional en conjunto con la información adquirida del entorno constituyen el **modelo operacional** sobre el cual se toman decisiones de control. El **modelo operacional**, dependiendo del tipo de control, tendría implementación computacional y representación virtual, y podría ser actualizado de forma interactiva, dependiendo de las capacidades de la interfaz.

Se podría definir **SA** (“Situational Awareness”) como el conocimiento más reciente del estado del entorno y el registro de éste en un **modelo**, junto con el conocimiento del estado operacional del dispositivo bajo control.

SA tendría distintos grados de formalidad, es decir los **modelos** podrían ser desde representaciones mentales hasta implementaciones computacionales formales. **SA** también está directamente relacionado con **telepresencia**; ya que sólo a través de la telepresencia, un operador en la estación de control construye (actualiza, conserva, mantiene) un mapa mental del estado del entorno remoto.

Pueden presentarse dos situaciones: en la primera, el modelo del entorno y el modelo del telerobot están presentes en la mente de un operador **entrenado** y experimentado; en el otro caso, los modelos operacionales (o partes de éstos) están presentes en un sistema computacional “experto” en la forma de bancos de datos, programas, algoritmos; modelos computacionales.

Si además, el modelo computacional cuenta con representación virtual, el tipo de control en donde se utilizaría sería **supervisado** y **semiautomático**, utilizando realidad aumentada para comunicar el estado del programa a un operador, para que éste lo actualice o supervise.

Este aspecto de la formalización del modelo es interesante, puesto que en todo el espectro de control, desde directo hasta supervisado, el **modelo** o partes de éste fluyen, se comunican, de la mente del operador a la computadora: el modelo del entorno se actualiza, formaliza o implementa, la carga de trabajo se delega o alivia; y de la computadora a la mente del operador: por medio de telepresencia, ambientes virtuales, realidad aumentada, y todo esto sucede a través de la **interfaz**

La capacidad de operadores, equipos de ingenieros y programadores de transcribir o codificar, es decir, formalizar esa elusiva forma abstracta e intuitiva que es el modelo mental del entorno y el telerobot, determina el nivel de automatización y el tipo de control del sistema bajo control, es decir, las tareas que podrán delegarse a la computadora por estar altamente modeladas. Esa forma abstracta y elusiva del **modelo** muchas veces carece de descripción en la mente del operador, pero se conoce de forma sensorial.

La **interfaz** facilita la comunicación entre computadora y humano. Por lo tanto, parte del problema de diseño de una interfaz pertenece a la psicología [16](Goodrich, p.216). Entre las funciones de una interfaz, está el facilitar el flujo de información entre humano y computadora, para delegar la carga de tareas y construir o valorar el SA.

La comunicación intuitiva para actualizar el modelo o corregirlo es lo que determina el grado de automatización de una aplicación. La frecuencia y eficiencia con la que el modelo virtual (implementado y formal) se actualice (o crezca) depende de las operaciones expuestas por la interfaz. La actualización comprende la traducción y formalización de un modelo informal (carga de trabajo) presente en la mente de un operador: tareas y características del entorno identificadas por el operador, por ejemplo la especificación de obstáculos y restricciones durante la generación de trayectorias. Si la interfaz no presenta operaciones expuestas, entonces solamente los diseñadores del sistema en conjunto con el operador entrenado deben concretizar el modelo mental en uno virtual en forma de ecuaciones, algoritmos, programas, y bases de datos. Pero esta forma de actualización del modelo no es interactiva, se realiza separada de la operación.

1.12 Incertidumbre

Con el propósito de que el operador reaccione de forma intuitiva ante eventos en el entorno remoto e identifique y seleccione objetivos y tareas, primero se debe reducir la **incertidumbre**, es decir inducir y aumentar el **conocimiento acerca del entorno**.

La **reducción de incertidumbre** es la tarea de más alta prioridad en un sistema **teleoperado**, es decir el registro y actualización continua del modelo virtual; o el aumento en **SA** dada su importancia. Por lo que son indispensables los sensores situacionales, y una **interfaz multimodal**. De aquí también se deriva la importancia de la inteligencia artificial, la visión artificial, etc.

La diferenciación, reconocimiento, clasificación, procesamiento, análisis, formalización, comunicación y transmisión de toda esta información reciente del entorno remoto y la integración en un mapa, pertenece al dominio de los problemas que son atacados con tecnologías de información: computación, algoritmos, etc. vital en un sistema teleoperado.

La **reducción de incertidumbre** comprende la **formalización** del **modelo** en una representación computacional o la **actualización** de alguno ya implementado (o la formalización teórica con propósitos de análisis y comunicación científica), es decir, comunicar o transcribir toda información y experiencia que se encuentran en la mente del operador en forma de **carga de trabajo** a la interfaz, con el fin de dividir la carga de trabajo entre el humano y la computadora. Como la memoria de un operador es limitada y éste puede perder la concentración ante cualquier interrupción, ésta forma mental del **modelo** es frágil (pues se requiere reconstruir por cada operación). Es por esto que la computadora puede extender, respaldar, delegar, intercambiar o sustituir las capacidades cognitivas de un operador, dando lugar a los distintos tipos de control supervisado

En la formalización influyen factores como la experiencia del operador; si el operador ha sido previamente entrenado, es decir, tiene un **modelo** a priori, teórico o formal, o si el **modelo** es práctico, empírico o informal. El operador **actualiza** y refina su modelo a priori con nuevas experiencias, y después actualiza el modelo computacional, si éste está disponible, y si la interfaz facilita la actualización.

1.13 Carga de trabajo de formalización

La **carga de trabajo** corresponde al conjunto de operaciones o tareas (funciones, objetivos, transiciones, restricciones, conocimiento del entorno y modelos) que se desean comunicar sobre la **interfaz** y ejecutar **sobre el entorno** a través del teleoperador.

Se requiere por lo tanto una previa reducción de incertidumbre sobre el entorno: un previo modelado o mapeo del entorno, formal o informal, pues toda tarea y objetivo está descrito en términos del entorno. Entonces, las primeras tareas consisten en obtener un mapa del entorno, a partir de telepresencia.

Cada tarea tiene prioridad y costos y **funciones objetivo** de acuerdo a un **modelo operacional**. La determinación de prioridad se basa en dicho **modelo**. Los costos pueden ser formalizados en un **modelo**. También hay costos de formalización e implementación [28](Sheridan, p.276).

Si la **carga de tarea** está presente en la mente del operador (incluyendo **SA**, valoración situacional y conocimiento del entorno), el **modelo operacional** (y las funciones objetivo) es informal; la **carga de tarea** tendrá un máximo. Si la computadora y el humano comparten la **carga de tareas**, se trata de **control compartido** o control **extendido**. Esto es posible si la **carga de trabajo** se formaliza; es decir, se implementa en un sistema computacional: se representa en forma de estructuras de datos, algoritmos y programas. La formalización entonces divide la carga de trabajo si se implementa: una carga estará presente en un sistema computacional "experto", y otra carga en la mente del operador. La formalización permite la comunicación, implementación, almacenamiento y división de **carga de trabajo**, análisis y optimización futuras, monitoreo, supervisión, etc., funciones que se sintetizan en una **interfaz de supervisión**, que facilitan la comunicación interactiva.

En toda forma de operación existe una **carga de trabajo**. En **control directo**, el modelo (programa) operacional es parte de la carga de trabajo. La carga de trabajo incluye las tareas relacionadas con determinar el estado situacional del dispositivo, a partir de información del entorno. Éstas tareas se consideran de alto nivel, quedan a cargo del operador y se suman a su **carga de trabajo**.

En **control supervisado** o **compartido** [28](Sheridan_92 p.65), existe un **modelo** implementado en la computadora, aunque el modelo siempre es limitado. Las **experiencias** se mantienen en la mente de un operador **entrenado**; sólo las **tareas** bien **modeladas** deben asignarse o comunicarse a la computadora, por medio de una interfaz, de forma interactiva, aunque el nivel y sofisticación del **modelo** de cada tarea puede incrementarse con el tiempo, y por lo tanto ser asignado y utilizado de forma automática por la computadora.

El equivalente de la **carga de trabajo** en la computadora, entonces, corresponde con algoritmos, programas y bases de datos

1.14 Telepresencia

Descrito por Sheridan como el "ideal de percepción de información y la **comunicación** con el operador humano de forma suficientemente natural de modo que se experimente la sensación de estar físicamente presente en el sitio remoto" [28](Sheridan_92 pp. 6,199).

Según Cui los sistemas de telepresencia están compuestos de tres partes: un sistema de captura para registrar y representar la información del sitio remoto, un sistema de transmisión, y un sistema de despliegue, para lograr que el usuario experimente la sensación de estar presente en la escena remota [11](Cui p.2).

La telepresencia se consigue con el gradual incremento de las capacidades sensoriales del teleoperador y las capacidades computacionales de la interfaz en la estación de control.

Telepresencia entonces, es un sistema remoto que combina visión computarizada y entornos virtuales, para conseguir una integración todavía más íntima con el operador humano. Con el propósito de identificación de

eventos y reacción ante éstos, y con el objetivo de que los **modelos** del entorno se incrementen, sean más nítidos en la mente del operador, explotando su memoria sensorial, en respuesta a la asociación de sonido e imágenes con estados operacionales y situacionales; y que los modelos operacionales sean generados y utilizados más intuitivamente y **actualizados** en un **mapa computacional**, aliviando la **carga de trabajo**.

El mapa computacional debe de ser interactivo puesto que debe ser utilizado durante la operación.

Por otra parte, **transparencia** es el grado de **telepresencia** que se ha alcanzado en el sistema teleoperado [6](Basañez p.454). Significa que la información adquirida en el sitio remoto es desplegada al operador en la misma naturaleza: temperatura en temperatura, presión en presión, tacto en tacto, fuerza en fuerza, etc.

En un sistema opaco es difícil de obtener SA. En contraste, un sistema completamente consciente del entorno, de todos los detalles sensoriales posibles o “mapeables” tendría **SA** completo, o **transparencia** en términos de Sheridan. Se necesita instrumentación especializada que está lejos de ser diseñada e implementada. Quizá solo debiera limitarse la **transparencia** al dominio de la tarea o aplicación.

Diversas técnicas de gráficos computacionales actuales como realidad aumentada (AR por sus siglas en inglés), o entornos virtuales (VE idem) sirven para aumentar el nivel de **telepresencia** (interactiva; operacional) en un sistema teleoperado e inducir **SA** en el operador. Con éstas técnicas se da representación visual a objetos e información abstracta o de distinta naturaleza

La realidad aumentada es una simulación o representación virtual o computarizada del entorno, donde se superpone información de alto nivel sobre la imagen. Esta superposición sirve para incrementar la calidad de la imagen, reducir el ancho de banda al utilizar modelos tridimensionales texturizados, combinando visión por computadora, etc.

La presentación de información al usuario depende de las capacidades de la interfaz que se esté utilizando. Existen dispositivos tácticos, acústicos y visuales. Los más desarrollados son los dispositivos visuales (“Darwinian vs Technological evolution” [29]). Una interfaz visual puede explotarse al máximo para presentar información de distinta naturaleza (táctil; térmica) al usuario. Esto se conoce como **visualización**.

Con la visualización de información, la comunicación es más eficiente, ya que el ancho de banda de la visión es muy grande, aunque requiere de atención continua. La información comunicada visualmente deberá ser interpretada e integrada en un **modelo** por el operador humano, y se espera que el operador reaccione inmediatamente ante nuevos eventos.

En resumen, telemetría, telefonía y televisión son canales de información provenientes del telerobot. **Telemetría** se refiere a variables operacionales internas; **telefonía** y **televisión** se refieren a variables situacionales o de entorno. Su integración se denomina **telepresencia**. Teleoperación es un canal de control hacia el telerobot. Un telerobot cierra sus propios ciclos de control y provee, además, funciones de telecomunicación y control multimodal

1.15 Interfaces

En toda telerobótica: desde control directo hasta control supervisado existen interfaces, [28](Sheridan_92 p.1) [13](Fong p.9) [16](Goodrich, p.216)

A la interfaz también se le llama panel de control, panel de instrumentos, dashboard, terminal, consola, etc.

Según Goodrich [16] hay tres tipos de interfaces de usuario: **HMI**, **HCI** y **HRI**

HMI: “Human-Machine Interface” o interfaz humano-máquina, son interfaces que se consideran locales a la máquina, ejemplos: SCADA, control de proceso.

HCI “Human-Computer Interface” o interfaz humano-computadora, se refiere solamente a interfaces e interacción con sistemas computacionales o de información. Los sistemas de información comprenden todo lo que son tecnologías de información (**IT**).

HRI. “Human-Robot Interface” o interfaz humano-robot, comprende las interfaces e interacción explícitamente con sistemas robóticos [16].

Las **HMI** podrían considerarse una abstracción que engloba todo un espectro de formas, desde las empuñaduras de herramientas sencillas hasta **HCI**.

Las **HCI** por su lado no necesariamente controlan una máquina, sino que en general son terminales que interactúan con sistemas de información e **IT**. Al estar separadas del proceso que controlan, las **HCI** pueden ser móviles.

En un sistema teleoperado, la interfaz podría considerarse **HMI**, por estar desprovista de un sistema computacional, y en telerobótica la interfaz podría considerarse **HCI** por estar integrada con un sistema computacional. Goodrich, [16] clasifica las interfaces de **teleoperación** como parte de la **HRI**

Las interfaces de **teleoperación** y las interfaces de alguna parte del espectro de control, se incluyen dentro de **HRI**. Se excluye de las de **HRI** a la de control automático o supervisado **de proceso** pues las interfaces para estas aplicaciones se consideran parte de **HMI**. Aunque en realidad la diferenciación de una interfaz podría depender de la escala del sistema analizada. Internet y las tecnologías de información permiten que las interfaces con sistemas computacionales sean utilizadas como **HMI** o **HRI**, y que además éstas estén separadas del proceso; permitiendo que sean móviles, y borrando la línea que distingue estas interfaces entre sí. Los requerimientos de alguna aplicación en particular, determinan la forma de la interfaz, por ejemplo, un paro de emergencia mecánico para **HMI**.

Las interfaces de **teleoperación** y control **supervisado** pertenecen al dominio de **HRI**, y requieren de interactividad además de funciones de alto nivel como diagnóstico, monitoreo, coordinación, integración, modelado y visualización de información del entorno etc.

El diagnóstico es la actividad de alto nivel equivalente a la determinación de errores en control automático. El diagnóstico se realiza por el operador; se alcanza por medio de **SA** y para ello se necesita **telepresencia**. El diagnóstico se basa en integrar información de distintas fuentes para determinar el estado operacional del telerobot y corregirlo si es necesario.

Para el diagnóstico es necesario el registro continuo de mediciones remotas de variables internas: telemetría. Es necesario un historial, un **modelo dinámico**, para comparar el estado actual con el histórico, etc.

Fong [13] divide las Interfaces de **teleoperación** en tres: **directa**, **multimodal** o multisensor, y **supervisión**

La **Interfaz directa** solamente cuenta con canales de video y con controles manuales (sin computadora)

En las Interfaces **multimodales** se integra un sistema computacional, y con ello se proveen una variedad de modos de control: actuador individual, control coordinado, control semiautomático; además de integración, modelado y visualización de información del entorno. Proveen de telepresencia, realidad virtual, entre otras utilidades. (“rich information feedback”)

Las **Interfaces de supervisión**, extienden las funciones provistas por la **interfaz multimodal**. Facilitan la generación de comandos de alto nivel y la actualización del modelo; poseen inteligencia artificial y proveen funciones de monitoreo, simulación, diagnóstico; añaden visualización de representaciones virtuales de las tareas a desempeñar. Se visualiza la carga de trabajo, además de información situacional integrada e información operacional, telemetría, etc. Poseen un sistema de telepresencia completo.

1.16 Diferencias entre operación y teleoperación

El prefijo griego **tele** dice muchas cosas:

La operación carece de canal de **telecomunicación**, es decir, la operación es **directa** sobre, e inseparable de la herramienta y la tarea; no puede extenderse, no es móvil.

En la operación la **interfaz** es la mano, y la empuñadura de la herramienta; El diseño de este tipo de interfaces se estudia en ergonomía y diseño industrial. Este tipo de interfaz se clasifica como **interfaz háptica**, debido a que sienten las fuerzas y pares directamente acoplados con la tarea. Las interfaces de operación incluyen desde empuñaduras no sofisticadas hasta las **HMI**, que se consideran locales a la máquina.

En la operación, el total de operaciones, funciones, tareas, objetivos, transiciones, restricciones, conocimiento del entorno y modelos se encuentran en la mente del operador como **carga de trabajo**.

Teleoperación requiere, además de un **telerobot** y un **enlace de telecomunicación**, una **interfaz multimodal** que provea **telepresencia**: una **interfaz** que decodifique e integre información remota situacional en un modelo virtual y represente de forma visual al operador; incluyendo un **sistema computacional** y **sensores situacionales**

1.17 Diferencias entre teleoperación y control supervisado

Si se corta la comunicación, y el control cesa, eso indica **teleoperación**. Si se corta la comunicación y el control u operación continúan, se trata de un **telerobot** y **control supervisado**, aunque un **telerobot** puede tener **modos** de operación tipo **control directo** en los que se somete completamente a las decisiones de control, estabilidad y coordinación de un operador o un equipo de operadores.

Un sistema **teleoperado** debe estar provisto de un **enlace de telecomunicación** (alámbrico o inalámbrico) con **terminales**; un puerto en el extremo maestro para presentar y comunicar efectiva e *intuitivamente* conocimiento acerca del entorno, y delegar el problema de **integración, mapeo, reducción de incertidumbre** y decisión al usuario u operador, es decir de generar o determinar **SA**, sea el sistema teleoperado autónomo o no, a fin de acoplar el entorno remoto y las tareas remotas con la inteligencia y voluntad del operador. Este acoplamiento también puede ser incrementado por la computadora, volviendo al control **semiautomático y compartido** [28].

En control supervisado se interactúa con información. El principal objetivo de un **sistema teleoperado** es la delegación de las tareas de integración y reconocimiento de información del entorno: aprovechar e integrar, la inteligencia humana y la precisión de una máquina, es decir **extender e integrar** las capacidades motrices, mecánicas y cognitivas de un ser humano y las de una máquina: precisión y fuerza, repetibilidad. Por lo que en el diseño del sistema intervienen ergonomía, psicología, etc. [16](Goodrich, p.216)

Funciones Interfaz Multimodal. Inducir SA, estado operacional actualizado y codificado continuamente por un humano en la estación de control o actualizado automáticamente en un modelo computacional; frecuencia de ciclo de control

1.18 Modelo de un sistema teleoperado.

A partir de la discusión anterior, se identifican las partes funcionales del sistema teleoperado: Un **enlace de telecomunicaciones**, un **sistema computacional**, una **interfaz multimodal**, un **sistema de telepresencia**, y un **teleoperador**.

El **teleoperador** a su vez es un sistema robótico que comprende sistemas de telemetría e instrumentación, sistema de control, sistema de potencia, servomecanismos, actuadores y sensores situacionales, como se menciona en [4](Barrientos, pp. 15-46)

El enlace de telecomunicaciones por su parte, comprende protocolos de comunicación, codificación y serialización del marco o cuadro de parámetros de estado (telemetría) y mediciones de entorno, y se divide en datos y control, dependiendo de la dirección del flujo de la información

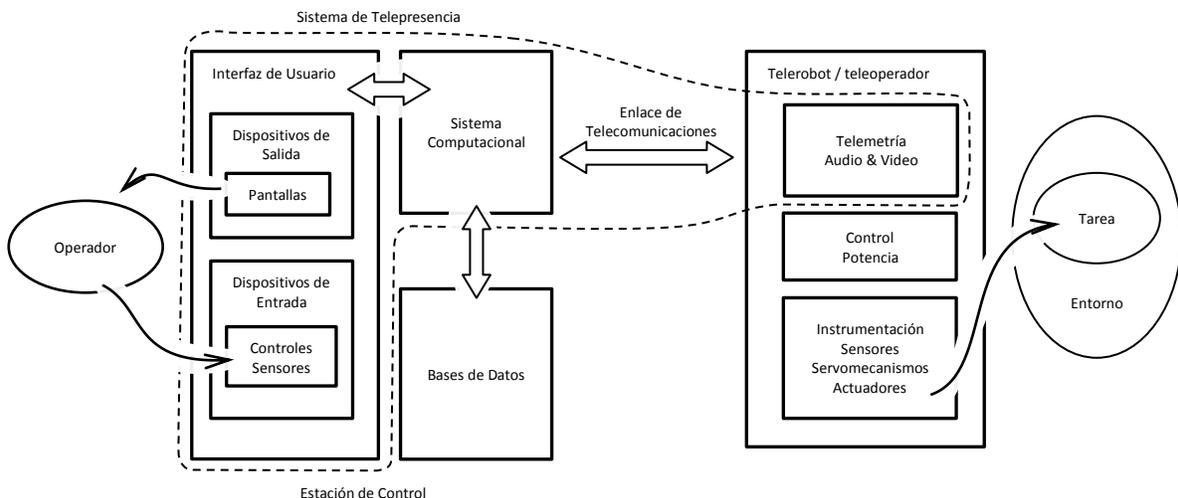


Figura 1 Flujos de información en el sistema teleoperado

Las funciones de “Extender, delegar, respaldar, intercambiar, sustituir” de la automatización segmentada del **espectro** de control que se menciona en [28](Sheridan_92, p. 3) discutido anteriormente, depende de los flujos de información presentes entre los distintos componentes, dando lugar a los distintos tipos de control: Directo, supervisado, semi-automático, compartido, cooperativo, etc. Si no hay componentes, tampoco hay flujo.

Como se puede observar, al telerobot se le añade un sistema de telepresencia que consiste en tres partes [11](Cui, p.2): Interfaz con sistema computacional, enlace telecomunicación, y sensores situacionales remotos

El sistema computacional finalmente, coordina y automatiza el resto de los componentes controlando el flujo de información entre éstos. Incluye software de operación, microcontroladores, bancos de datos, inteligencia artificial, etc.

En [16](Goodrich, p. 247) se mencionan campos científicos relacionados con el sistema teleoperado. En resumen, un robot es un sistema mecatrónico con inteligencia; un telerobot es un robot con telecomunicaciones y estación de control con telepresencia

1.19 Aplicaciones en teleoperación.

En general las aplicaciones consisten en navegación, monitoreo y manipulación en entornos remotos o peligrosos, o en ambientes no estructurados. Se presenta a continuación una tabla con las aplicaciones más comunes. Estas aplicaciones son clasificadas como pertenecientes al sector de “robots de servicio”

Tabla 1 Campos de aplicación en teleoperación

1	Industrial (SCADA) y construcción, industria de generación eléctrica, industria nuclear [6]
2	Agricultura
3	Minería y excavación [6](Basañez p.459) [24]
4	Científico: control, automatización e instrumentación de experimentos. (Control de proceso científico) [6]
5	Navegación de robots móviles [13] [21]
6	Robots exploradores, ej.: Robot explorador “Dante II” [3]
7	Oceanografía, Exploración submarina y marítima (ROVs Remote Operated Vehicles) [40], [39], [15], [11].
8	Aeroespacial: Aviación comercial, Satélites y telecomunicaciones, Exploración Planetaria [28],[29],[27],[11]
9	Medicina: cirugía telerobótica. Medical Robotics & "MIRS" (por las siglas en ingles de: Minimally Invasive Robotic Surgery) [36] [2] [38] [12] [11]
10	Transporte [16](Goodrich p.251)
11	Entretenimiento
12	Militar y Defensa, Seguridad, Vigilancia [39]
13	Servicio, limpieza, manejo, recuperación y manipulación de desechos y materiales peligrosos o tóxicos (por ejemplo en medicina) [14] [6]
14	Inspección y Mantenimiento [39] [6]

1.20 Descripción del proyecto.

División del proyecto de teleoperación.

Un proyecto de teleoperación se podría enfocar a diseño de cualquier elemento funcional del **modelo de teleoperación**: Diseño mecatrónico mecánico del manipulador, diseño de sistema de servocontrol,

infraestructura de instrumentación; infraestructura de telecomunicación y telemetría, protocolos de comunicación, sensores situacionales, sistema de telepresencia, diseño de plataforma virtual, interfaz multimodal, inteligencia artificial, visión por computadora, sistemas computacionales, bases de datos, etc. Se disponía en un principio únicamente con un prototipo de un manipulador robótico de cinco grados de libertad con estructura mecánica hecha de partes de acrílico cortadas por láser, y actuado por servomotores de aerodelismo. No existía comunicación entre el prototipo y la computadora. Dada la complejidad de un proyecto de estas características, se escogió por enfocarse a lo básico: implementación del enlace de telecomunicación y control por computadora.

1.21 Objetivos

- 1 Teleoperar al manipulador: Alcanzar el movimiento interactivo, coordinado y semiautomático (teleoperado; en tiempo real; en línea) de las articulaciones del robot prototipo (funciones y requerimientos de control supervisado), y monitorear el desempeño.
- 2 Construir un entorno de realidad virtual que sea capaz de mostrar un modelo tridimensional de un manipulador de cinco grados de libertad y demostrar que con el entorno se es capaz de interactuar y transferir comandos de teleoperación al robot prototipo, y también: monitorear, resolver la cinemática inversa, planeación de movimiento, crear algoritmos de evasión de obstáculos, etc. para un manipulador físico prototipo de cinco grados de libertad.
- 3 Demostrar control supervisado y la delegación de tareas a la computadora por medio de la actualización interactiva de la implementación computacional del mapa del entorno y la ejecución semiautomática de evasión de obstáculos.

1.22 Metodología

- 1 Resolver e implementar las funciones de control cinemático inverso para un robot en configuración serial de seis grados de libertad.
- 2 Programar el microcontrolador esclavo encargado del control (actuación) de los eslabones del prototipo, también encargado de la comunicación bidireccional coordinada por puerto serial con la computadora;
- 3 Implementar y depurar el protocolo de comunicación entre el robot prototipo y la computadora
- 4 Experimentar con la ejecución de teleoperación del dispositivo y la actualización del modelo computacional.
- 5 Depurar el sistema.

1.23 Descripción del experimento.

1. Comprobación numérica de la cinemática directa. Comparar la sustitución directa de parámetros articulares en la fórmula obtenida con la multiplicación numérica de matrices de postura.
2. Comprobar en conjunto el funcionamiento de los distintos dispositivos, componentes, y programas implementados, utilizando como prueba la ejecución de algoritmos de control cinemático sencillos, específicamente evasión de obstáculos, navegación en espacio de trabajo y manipulación de objetos.
3. Demostrar la actualización del modelo computacional.

Análisis Cinemático del Manipulador

2.1 Representaciones de la orientación

Existen diversas formas de representar la orientación de un cuerpo rígido en tres dimensiones. Se discuten dos: matrices de rotación, y ángulos de Euler. Las matrices de rotación, forman parte de la matriz homogénea para transformaciones afines. Con la matriz homogénea es computacionalmente más sencillo realizar transformaciones geométricas en el espacio tridimensional. Los ángulos de Euler sirven para determinar una orientación específica, y a partir de éstos se obtiene una matriz

Existen otras representaciones útiles como los cuaterniones [31], [25], que facilitan el cálculo de la interpolación de orientaciones, en el caso de seguimiento de trayectorias en \mathbb{R}^6 , problema que pertenece al control cinemático.

2.2 Matrices de rotación

Son matrices (subconjunto del conjunto Especial Ortogonal [31](Shoemaker p. 246) que representan una rotación respecto de un eje arbitrario [10](Craig, p.20), [33](Spong, p.32), [4](Barrientos, p.53). Toda rotación arbitraria puede descomponerse en rotaciones elementales llevadas a cabo con respecto a los ejes coordenados. El sentido positivo de una rotación se toma como la dirección del eje coordenado.

La matriz de rotación transforma las coordenadas (componentes) de un vector tridimensional de tal forma que la resultante es un vector rotado.

Se supone que la matriz de rotación representa la orientación de algún marco de referencia arbitrario. Todo vector descrito desde el marco de referencia arbitrario se describe a partir de un conjunto ortonormal de vectores $\{u, v, w\}$ (mutuamente perpendiculares y de longitud unitaria). Se dice que estos vectores forman una base para este marco.

El problema de obtener la versión rotada o reorientada p' de un vector p , consiste en encontrar las componentes $\{x', y', z'\}$ cuya combinación lineal en la base $\{u, v, w\}$ del marco de referencia arbitrario coincide con la descripción $\{x, y, z\}$ del vector original.

La matriz de rotación se obtiene especificando las coordenadas de esta base en términos del marco de referencia fijo. Si el marco de referencia fijo es $\{x, y, z\}$, se obtienen las componentes u_x, u_y, u_z del vector u ; las componentes v_x, v_y, v_z del vector v y las componentes w_x, w_y, w_z del vector w .

Tratándose de vectores columna, estas componentes se ordenan como sigue:

$$\begin{bmatrix} u_x & v_x & w_x \\ u_y & v_y & w_y \\ u_z & v_z & w_z \end{bmatrix}$$

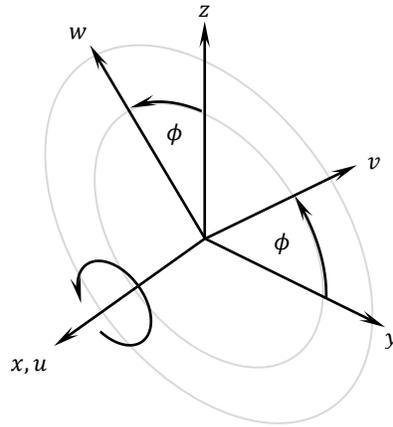


Figura 2.1 Rotación elemental en x

Observando la figura 2.1, se encuentran las componentes del sistema rotado sobre el eje x. Agrupando en una matriz se obtiene:

$$R(x, \phi) = R_x(\phi)$$

$$R(x, \phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix}$$

Que es la matriz de rotación elemental sobre el eje x

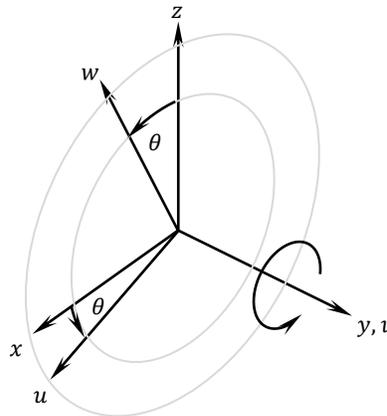


Figura 2.2. Rotación elemental en "y"

Del mismo modo, obtenemos la matriz de rotación elemental sobre el eje y:

$$R(y, \theta) = R_y(\theta)$$

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

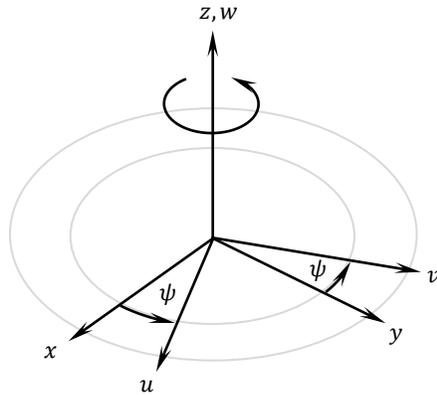


Figura 2.3. Rotación elemental en z

Por último, de la figura 2.3:

$$R(z, \psi) = R_z(\psi)$$

$$R(z, \psi) = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2.3 Ángulos de Euler

Los ángulos de Euler introducidos por el matemático suizo Leonhard Euler en mecánica de cuerpo rígido, constituyen un conjunto de tres coordenadas angulares que sirven para especificar la orientación de un sistema de referencia cualquiera respecto de un marco considerado fijo. Sea "A" ($\{S_A\}: X_A Y_A Z_A$) el sistema de referencia. Sea "B" ($\{S_B\}: X_B Y_B Z_B$) un sistema con orientación arbitraria. Se desea conocer la orientación de "B" respecto de "A". Lo que se busca es alguna secuencia de rotaciones elementales que se tienen que realizar sobre "A" para obtener "B", ó alinear "A" con "B". Una secuencia en particular describe la orientación de "B" respecto de "A" en "coordenadas angulares". Existen 12 posibilidades diferentes [1] (Arking p.491): cualquier secuencia de tres ejes tales que dos ejes consecutivos no sean los mismos. Se denominan convenciones y se dividen en dos grupos: clásicos y náuticos. Los ángulos de Euler "propios" o "clásicos" forman el primer grupo. Repiten un eje para realizar la última rotación, (xyx, xzx, yxy, yzy, zxz, zyz). Los ángulos, "náuticos", "Tait-Bryan" o "de Cardan" forman el segundo grupo; las rotaciones se realizan en los tres ejes disponibles. (xyz, xzy, yxz, yzx, zxy, zyx). [10](Craig p.41), [33](Spong, p.44), [4](Barrientos, p.57)

Pueden ocurrir situaciones en las que la descripción de la orientación de "B" respecto de "A" sea la misma, a pesar de que "B" permanezca con orientación constante respecto de un marco inercial "C". Sucede cuando se considera que el sistema "A" no necesariamente se encuentra fijo; bien puede estar sujeto a una rotación y tener una orientación variable respecto del tercer sistema "C"; Entonces siempre existen al menos dos posibles alineaciones geométricas que describen la orientación de "B" respecto de "A": "Alibi", cuando "B" se orienta respecto de "A" y "Alias", rotación del sistema de referencia "A", cuando "B" permanece fijo y "A" se reorienta. En la discusión y análisis del resto del capítulo se consideraran transformaciones de tipo alibi. Es decir el marco de referencia se considera fijo.

La selección de ejes coordenados depende del uso que se le quieran dar, por ejemplo en navegación y aeronáutica, la selección más frecuente es X para la dirección de movimiento, Z para la vertical local y Y completa un sistema de mano derecha.

Diferente selección de ejes da lugar a una distinta convención (secuencia de rotaciones elementales a seguir) de ángulos de Euler. Una misma convención describe orientaciones diferentes si es utilizada en sistemas coordenados diferentes. Además, los nombres de las rotaciones y los símbolos con que se representan, dependen del gusto particular de cada autor.

Entonces, como toda orientación siempre va referida a un sistema de coordenadas particular, éste debe especificarse al inicio de la discusión.

La definición geométrica, se basa en escoger dos planos, uno en el sistema de referencia y otro en el sistema móvil. A cada plano le corresponde un eje normal y viceversa

Al eje X_A le corresponde el plano normal: " $Y_A Z_A$ ", al eje Y_A le corresponde el plano $X_A Z_A$, etc.

Se dice en este contexto que dos planos son homólogos cuando los ejes normales a éstos son equivalentes entre sí; por ejemplo, son planos homólogos los planos ortogonales a los ejes Z_A y Z_B

Cuando los planos seleccionados no son homólogos, por ejemplo, los planos correspondientes a Y_A y Z_B , entonces la convención se conoce como Tait-Bryan.

La intersección de los planos escogidos se denomina como "línea de nodos". Es a partir de esta línea que se definen las tres coordenadas angulares.

La rotación de la proyección ortogonal de un punto sobre una línea, genera un disco contenido en un plano perpendicular a dicha línea. Si consideramos las normales de dos planos como ejes de rotación, la línea de intersección de los planos, es aquella única rotación común entre ambos ejes.

Para el caso Z-X'-Y''

α es el ángulo entre el eje X_A y la línea de nodos;

β es el ángulo entre el eje Z_A y el Z_B

γ es el ángulo entre la línea de nodos y el eje X_B

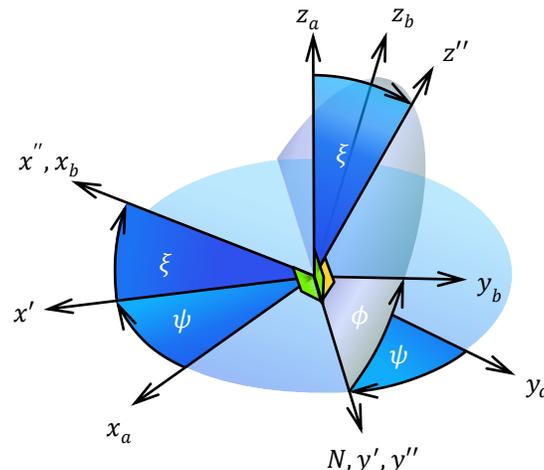


Figura 2.4 ángulos náuticos

En este trabajo se utilizará la notación (α, β, γ) para denotar ángulos de Euler propios y (ψ, θ, ϕ) para denotar ángulos náuticos (Tait-Bryan). Sólo en casos en que exista ambigüedad, se utilizará la letra griega "ksi": ξ para reemplazar θ

Una rotación intrínseca representa rotaciones en alguno de los ejes del marco móvil, mientras que una rotación extrínseca representa rotaciones en el marco de referencia fijo.

Una secuencia de rotaciones intrínseca X-Y'-Z'' produce el mismo resultado que una secuencia de rotaciones extrínseca invertida Z-Y-X, así que existen dos nombres posibles para cualquier convención, la forma intrínseca o la forma extrínseca. Se distingue un caso del otro por la presencia de superíndices romanos consecutivos en la letra correspondiente al eje coordenado. El orden de las rotaciones para el caso intrínseco X-Y'-Z'' sería: primero rotar el marco de referencia móvil alrededor de su eje X, luego rotar alrededor del eje móvil Y, y luego realizar la última rotación alrededor del eje móvil Z. Para el caso extrínseco, el orden de las rotaciones equivalente a la misma orientación sería: primero rotar el marco móvil

alrededor del eje fijo Z, luego rotar el marco móvil alrededor del eje fijo Y, y por último, rotar el marco móvil alrededor del eje fijo X

Esta noción es de importancia ya que el orden en que se realicen las multiplicaciones matriciales depende de si se trata de una secuencia intrínseca o extrínseca

Ocurren singularidades cuando ejes homólogos en el sistema de referencia y el sistema móvil se encuentran alineados; la intersección de los respectivos planos que da lugar a la línea de nodos, no queda definida. Esto se conoce como “gimbal-lock”

La matriz resultante para la convención náutica Z-Y'-X'', queda:

$$\begin{aligned}
 & R_z(\psi) R_y(\theta) R_x(\phi) \\
 &= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \\
 &= \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \phi & -\sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix} \\
 &= \begin{bmatrix} \cos \psi \cos \theta & \cos \psi \sin \theta \sin \phi - \sin \psi \cos \phi & \cos \psi \sin \theta \cos \phi + \sin \psi \sin \phi \\ \sin \psi \cos \theta & \sin \psi \sin \theta \sin \phi + \cos \psi \cos \phi & \sin \psi \sin \theta \cos \phi - \cos \psi \sin \phi \\ -\sin \theta & \cos \theta \sin \phi & \cos \theta \cos \phi \end{bmatrix}
 \end{aligned}$$

Y utilizando notación abreviada:

$$= \begin{bmatrix} c_\psi c_\theta & (c_\psi s_\theta s_\phi - s_\psi c_\phi) & (c_\psi s_\theta c_\phi + s_\psi s_\phi) \\ s_\psi c_\theta & (s_\psi s_\theta s_\phi + c_\psi c_\phi) & (s_\psi s_\theta c_\phi - c_\psi s_\phi) \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix}$$

Por último, se adjunta la tabla 2.1 de sinónimos para ángulos náuticos:

Tabla 2.1 Sinónimos para ángulos náuticos

Nombre eje	vertical	transversal	longitudinal
Símbolo eje	z	y	x
Símbolo ángulo	ψ	θ	ϕ
Nombre Inglés	Yaw	Pitch	Roll
Sinónimo Inglés	Heading	Elevation	Bank
Nombre español	Guiñada	Cabeceo	Alabeo
Sinónimo español	Deriva	Inclinación	

2.4 Determinación de la orientación náutica Z-Y'-X'' de un sistema de coordenadas B, descrito desde otro A

Se desea conocer en coordenadas angulares náuticas Z-Y'-X'' (Tait-Bryan, 2a convención de Euler: yaw, pitch, roll) la orientación de B relativa al sistema A cuando A y B tienen orientaciones arbitrarias. Esta es una operación que se realiza constantemente en la solución de cinemática inversa de una cadena cinemática en un robot.

Los pasos son los siguientes:

Se llama a la base de B: $\{x_b, y_b, z_b\}$ (o simplemente u, v, w), y a la base de A, $\{x_a, y_a, z_a\}$.

La definición de la orientación intrínseca es la siguiente:

Existe una rotación en z_a (de ψ radianes) que transforma el sistema $x_a y_a z_a$ en $x' y' z'$

Existe una rotación en y' (de θ radianes) que transforma el sistema $x' y' z'$ en $x'' y'' z''$

Existe una rotación en x'' (de ϕ radianes) que transforma el sistema $x'' y'' z''$ en $x_b y_b z_b$

Éstas son las rotaciones elementales en el marco de referencia móvil que alinean el sistema A con el sistema B.

Primero: se observa la matriz de rotación de cada sistema.

$$R_A = \begin{bmatrix} x_{ax} & y_{ax} & z_{ax} \\ x_{ay} & y_{ay} & z_{ay} \\ x_{az} & y_{az} & z_{az} \end{bmatrix}$$

$$R_B = \begin{bmatrix} x_{bx} & y_{bx} & z_{bx} \\ x_{by} & y_{by} & z_{by} \\ x_{bz} & y_{bz} & z_{bz} \end{bmatrix}$$

La primer columna de R_A corresponde a las componentes x_{ax}, x_{ay}, x_{az} de x_a . La tercer columna de R_B corresponde a las componentes z_{bx}, z_{by}, z_{bz} de z_b .

Segundo: Se determina la línea de nodos. Sea N la línea de nodos. N corresponde al segundo eje de rotación: y' . y' es perpendicular a z_a : por ser z_a paralela a z' , y por ser z' perpendicular a y' , ya que $x' y' z'$ completan una terna ortonormal. y' también es paralelo a y'' , por ser y' eje de rotación. Esto implica además, que y' es perpendicular a x'' ; ahora, siendo x'' paralelo al eje x_b , y x'' perpendicular a y' , implica que y' también es perpendicular a x_b . Entonces, como y' es perpendicular tanto a z_a como a x_b , indica que es la línea de nodos (ver Figura 2.5).

El eje y' puede entonces obtenerse mediante el producto cruz de z_a con x_b :

$$y' = N = z_a \times x_b$$

Tercero: se determina del el eje z'' (z' es paralelo a z_a), a partir del cual se mide el roll " ϕ " y el pitch " θ ". El eje z'' es el resultado de rotar z' alrededor de la línea de nodos (y') una cantidad de θ . Dicho de otra forma, el sistema $x'' y'' z''$ se obtiene de rotar el sistema $x' y' z'$ alrededor de y' . Por lo tanto, z' y z'' son perpendiculares a y' (ver figura 2.3.2)

z'' también es perpendicular a x_b por ser x_b paralelo a x'' .

Entonces z'' se determina a partir de x_b y la línea de nodos y'

Cuarto: se determina x' . x' forma una terna ortonormal con y' y z' ; z' es paralelo a z_a , entonces: x' es perpendicular tanto a la línea de nodos y' como a z_a (ver figura 2.6). El eje x' sirve para determinar el sentido del pitch θ . Si z'' tiene componente positiva sobre x' , el pitch es negativo. De forma similar se determina el sentido para yaw ψ , y roll ϕ .

Las coordenadas angulares (ψ, θ, ϕ) yaw pitch y roll, se determinan como sigue:

Yaw, denotado por ψ es el ángulo entre y_a y la línea de nodos y' ; ψ también es el ángulo entre x_a y x' (ver figura 2.5).

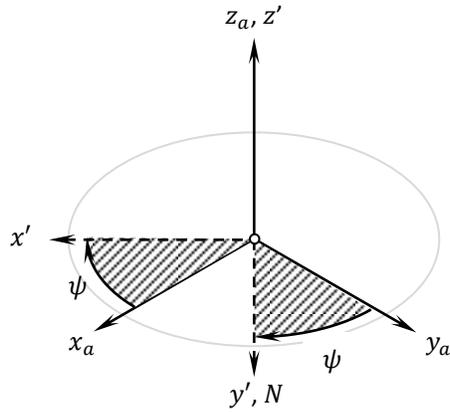


Figura 2.5 Obtención del yaw

Pitch, denotado por θ es por definición el ángulo entre z' y z'' . θ también es el ángulo entre z_a y z'' por ser z_a paralelo a z' ; y θ es también el ángulo entre x' y x'' (ver figura 2.6).

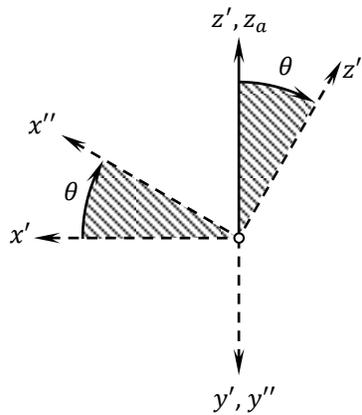


Figura 2.6 Obtención del pitch

Roll, denotado ϕ es el ángulo entre z'' y z_b ; ϕ también es el ángulo entre la línea de nodos y' y y_b (ver figura 2.7)

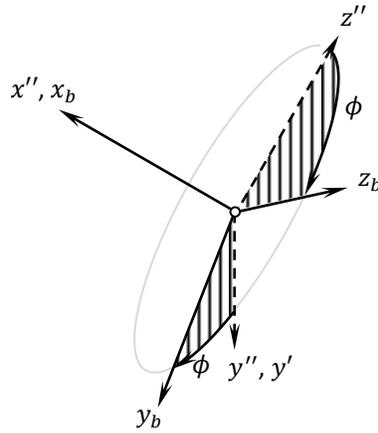


Figura 2.7 Obtención de roll

En resumen, los pasos son (ver figura 2.8):

Se calcula $N = y' = z_a \times x_b$

Se calcula $L = z'' = x_b \times N$

Se calcula $x' = N \times z_a$

Se calcula $\phi = \arccos(z'' \cdot z_b)$;

Si $(z_b \cdot y') > 0$, cambiar el signo de roll: $\phi *= -1$;

Se calcula $\theta = \arccos(z'' \cdot z_a)$;

Si $(z'' \cdot x') < 0$, cambiar el signo del pitch: $\theta *= -1$;

Se calcula $\psi = \arccos(x_a \cdot x')$;

Si $(x' \cdot y_a) < 0$, cambiar el signo del yaw: $\psi *= -1$;

Ó también:

$\psi = \arccos(y_a \cdot y')$; Si $(y' \cdot x_a) < 0$, $\psi *= -1$;

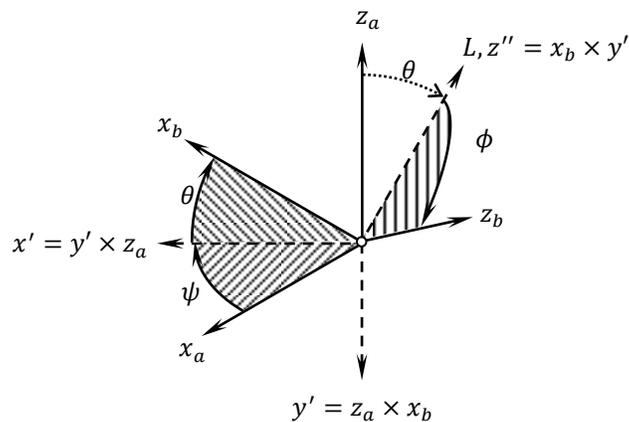


Figura 2.8 Secuencia completa de ángulos náuticos

2.5 Modelo Cinemático

El propósito de la cinemática directa es describir la postura del elemento terminal del robot, como función de los parámetros articulares de éste. Al elemento terminal o último eslabón del robot se le asocia un punto de operación y un marco de referencia. La postura comprende posición y orientación; al conjunto de posturas de algún conjunto de eslabones contiguos se denomina configuración. En el procedimiento de la determinación de postura del elemento terminal, se determinan también la posición y orientación de cada una de las articulaciones y eslabones restantes, especificando en todo instante la configuración geométrica del robot en el espacio. Si éste modelo cinemático se deriva, se obtienen las velocidades, y a partir de las derivadas de estas últimas las fuerzas y reacciones presentes en los eslabones. Las ecuaciones obtenidas del modelo sirven en la implementación de algoritmos de control dinámico.[25]

En el modelo cinemático se buscan ecuaciones de la siguiente forma:

$$x = f(q)$$

que relacionan la posición y orientación náutica codificadas en un vector de seis dimensiones x respecto al vector de parámetros angulares q .

La obtención de estas ecuaciones se basa en el método Dénavit-Hartenberg, sintetizado de [4], [33] y [10]. Consiste en describir la posición y orientación entre dos eslabones contiguos utilizando cuatro parámetros: α , a , d , θ ; por grado de libertad, que se obtienen como sigue: se especifican los ejes de rotación z_i , para cada eje o grado de libertad o articulación actuada, después se sitúan los sistemas S_i en la intersección del eje z_i con la línea normal común a z_{i-1} y z_i . El eje x_i se sitúa en esta línea. θ_i es el ángulo que hay que girar en torno a z_{i-1} para que x_{i-1} y x_i queden paralelos. d_i es la distancia que hay que desplazar S_{i-1} a lo largo de z_i para que x_{i-1} y x_i queden alineados. a_i es la distancia en x_{i-1} para hacer coincidir el origen de S_{i-1} con el origen de S_i . α_i es el ángulo que hay que girar en torno de x_{i-1} para que z_{i-1} y z_i queden paralelos.

En la figura 2.9 se observa la secuencia individual de transformaciones geométricas para describir la postura de un marco de referencia relativo a otro. En la figura 2.10, se observa la transformación total

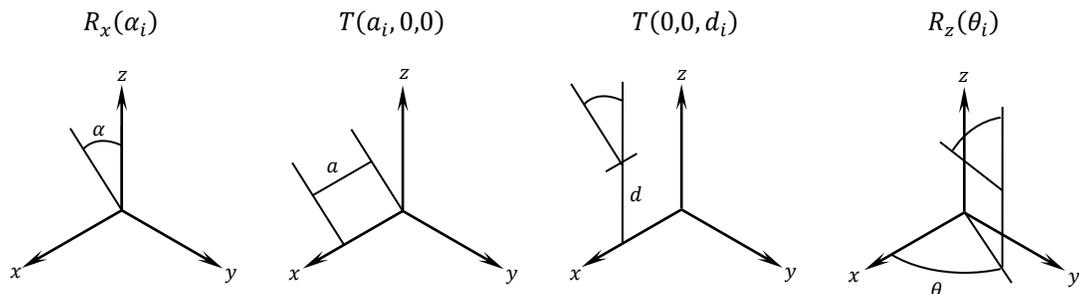


Figura 2.9 Secuencia de transformaciones DH

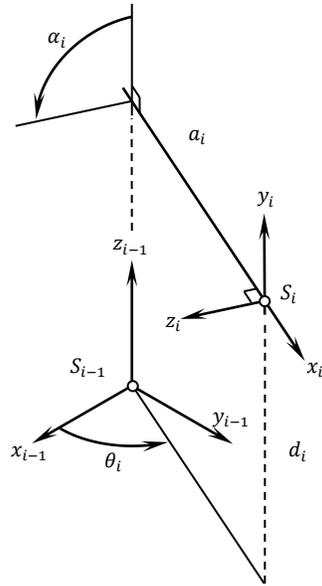


Figura 2.10 Secuencia completa DH

La matriz de transformación homogénea que representa la posición y orientación relativa entre dos eslabones sucesivos en un robot viene dada por: [4](Barrientos, p. 96), [33](Spong, p.65) [10](Craig, p.75)

$${}^{i-1}\mathbf{A}_i = R_z(\theta_i)T(0,0,d_i)T(a_i,0,0)R_x(\alpha_i)$$

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

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

$${}^{i-1}\mathbf{A}_i = \begin{bmatrix} {}^{i-1}\mathbf{R}_i & {}^{i-1}\mathbf{p}_i \\ \mathbf{0} & 1 \end{bmatrix}$$

En el robot prototipo se identifican cinco juntas rotacionales. Sin embargo se considera una más para que el análisis sea más completo, para un total de seis grados de libertad. z_3 es un eje virtual (ver figura 2.11). La dirección de los ejes z determina el sentido positivo de rotación. Esta elección es arbitraria, por lo que existen $2^6 = 64$ posibilidades diferentes.

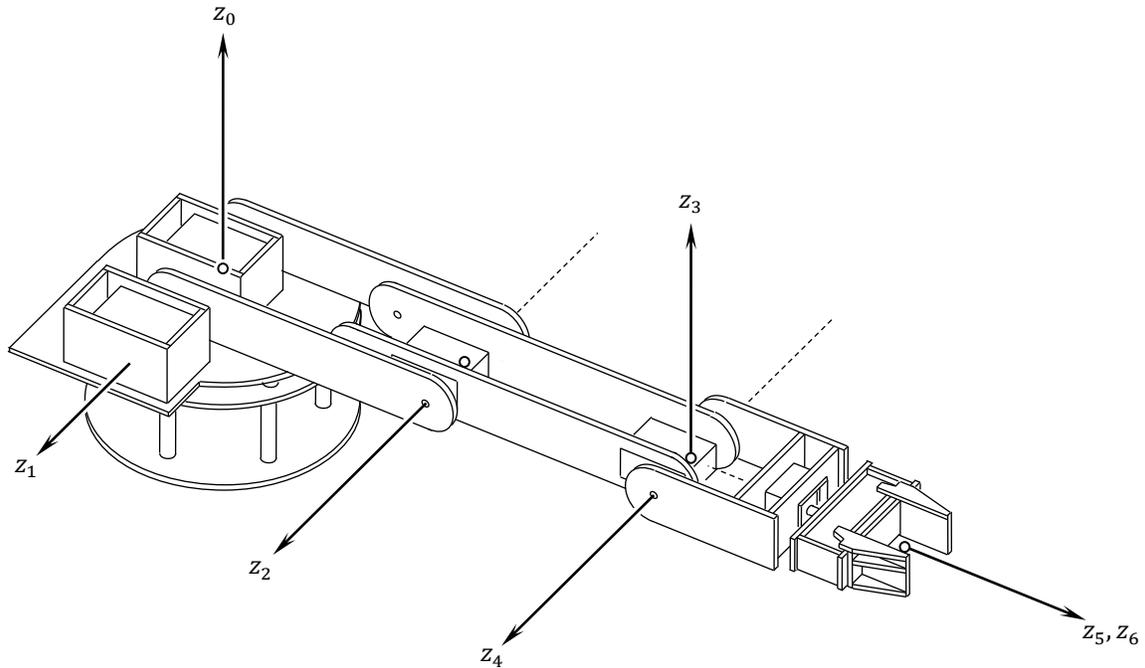


Figura 2.11 Selección de ejes z_i

A partir de los ejes z_i , se determinan los ejes x_i , como se observa en la figura 2.12

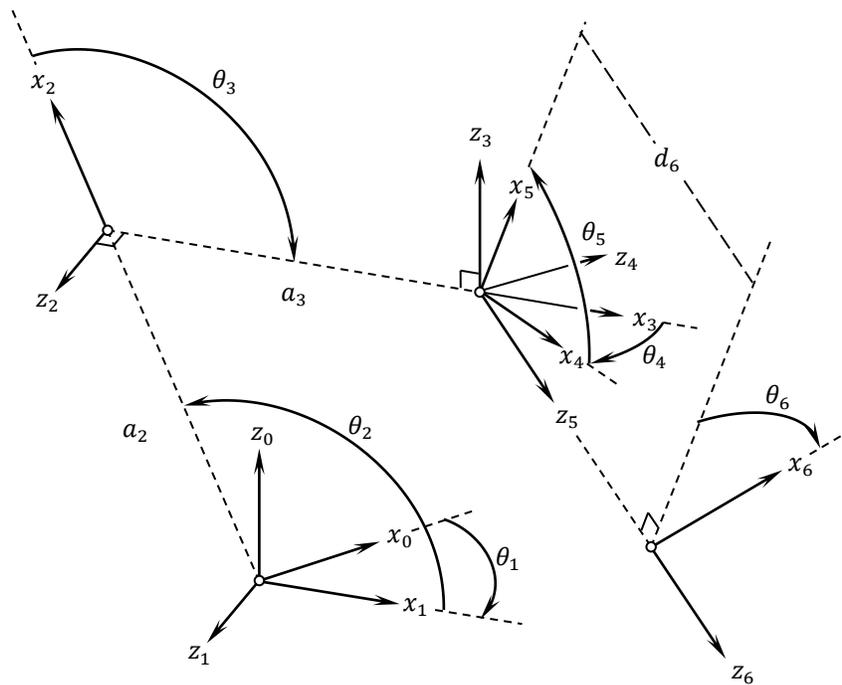


Figura 2.12 Obtención de los ejes x_i

En posición cero, el vector de parámetros (coordenadas) articulares $\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6)$, $\mathbf{q} = (\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$, $\mathbf{q} = (0,0,0,0,0,0)$ lo que implica que todos los ejes x_i son paralelos. En esta configuración se aprecian mejor los parámetros α_i , como se observa en la figura 2.13

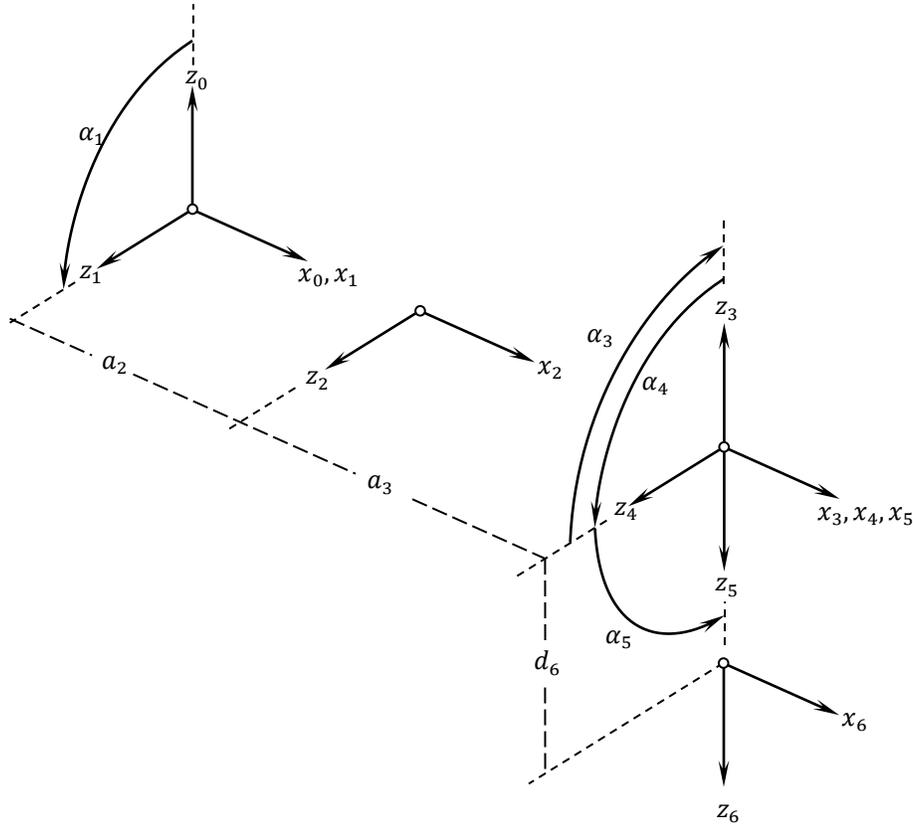


Figura 2.13 Determinación de los ángulos α_i

A partir de las figuras, 2.12 y 2.13 se determina la tabla 2.2:

Tabla 2.2				
i	θ_i	d_i	a_i	α_i
1	θ_1	0	0	90°
2	θ_2	0	a_2	0
3	θ_3	0	a_3	-90°
4	θ_4	0	0	90°
5	θ_5	0	0	90°
6	θ_6	d_6	0	0

A partir de los valores de la tabla 2.2, y la fórmula para la transformación homogénea entre eslabones se obtienen las siguientes matrices:

$${}^0A_1 = \begin{bmatrix} {}^0R_1 & {}^0p_1 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c_1 & 0 & s_1 & 0 \\ s_1 & 0 & -c_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1A_2 = \begin{bmatrix} {}^1R_2 & {}^1p_2 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c_2 & -s_2 & 0 & a_2c_2 \\ s_2 & c_2 & 0 & a_2s_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2A_3 = \begin{bmatrix} {}^2R_3 & {}^2p_3 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c_3 & 0 & -s_3 & a_3c_3 \\ s_3 & 0 & c_3 & a_3s_3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3A_4 = \begin{bmatrix} {}^3R_4 & {}^3p_4 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c_4 & 0 & s_4 & 0 \\ s_4 & 0 & -c_4 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4A_5 = \begin{bmatrix} {}^4R_5 & {}^4p_5 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c_5 & 0 & s_5 & 0 \\ s_5 & 0 & -c_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^5A_6 = \begin{bmatrix} {}^5R_6 & {}^5p_6 \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} c_6 & -s_6 & 0 & 0 \\ s_6 & c_6 & 0 & 0 \\ 0 & 0 & 1 & d_6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La transformación absoluta de la trama herramienta 0T_6 será entonces:

$${}^0T_6 = {}^0A_1 {}^1A_2 {}^2A_3 {}^3A_4 {}^4A_5 {}^5A_6$$

Desarrollando:

$${}^0T_6 = \begin{bmatrix} {}^0R_1 & {}^0p_1 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^1R_2 & {}^1p_2 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^2R_3 & {}^2p_3 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^3R_4 & {}^3p_4 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^4R_5 & {}^4p_5 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} {}^5R_6 & {}^5p_6 \\ \mathbf{0} & 1 \end{bmatrix}$$

Esta forma es computacionalmente eficiente, sin embargo pierde claridad al efectuar las operaciones matriciales para obtener la forma algebraica. (que se utilizará en la verificación experimental de cinemática directa comparándola con el método numérico)

Para facilitar el seguimiento del desarrollo de 0T_6 , se calculan por separado los valores de la submatriz de rotación y los valores del subvector de posición. (ver apéndice 6.1).

Una vez obtenida la matriz de la trama herramienta, y usando la nomenclatura de [4](Barrientos p. 94), se buscan ecuaciones de la forma:

$$\mathbf{x} = (x, y, z, \psi, \xi, \phi) = \vec{f}(\mathbf{q})$$

Dónde:

$$\begin{aligned} x &= f_x(q_1, q_2, q_3, q_4, q_5, q_6) = f_x(\mathbf{q}) \\ y &= f_y(q_1, q_2, q_3, q_4, q_5, q_6) = f_y(\mathbf{q}) \\ z &= f_z(q_1, q_2, q_3, q_4, q_5, q_6) = f_z(\mathbf{q}) \\ \psi &= f_\psi(q_1, q_2, q_3, q_4, q_5, q_6) = f_\psi(\mathbf{q}) \\ \xi &= f_\theta(q_1, q_2, q_3, q_4, q_5, q_6) = f_\xi(\mathbf{q}) \\ \phi &= f_\phi(q_1, q_2, q_3, q_4, q_5, q_6) = f_\phi(\mathbf{q}) \end{aligned}$$

Las coordenadas x, y, z representan la posición del punto de operación P , y (ψ, ξ, ϕ) la orientación náutica (Tait-Bryan Z-Y'-X'') del extremo del robot, respecto del sistema base S_0 en función de las coordenadas articulares q_i .

La matriz 0T_6 indica la localización del sistema de coordenadas asociado al extremo del robot (trama herramienta) con respecto al sistema de referencia de la base del robot, en términos de los parámetros articulares. Cada elemento de esta matriz es igual a términos trascendentes

En [33](Spong, p. 70) se da una nomenclatura para la trama herramienta utilizando n, s, a , que se utilizará en adelante:

"The unit vectors along the x_n, y_n , and z_n axes are labeled as \mathbf{n}, \mathbf{s} , and \mathbf{a} , respectively. The terminology arises from fact that the direction \mathbf{a} is the approach direction, in the sense that the gripper typically approaches an object along the \mathbf{a} direction. Similarly the \mathbf{s} direction is the sliding direction, the direction along which the fingers of the gripper slide to open and close, and \mathbf{n} is the direction normal to the plane formed by \mathbf{a} and \mathbf{s} "

Entonces, \mathbf{n} es el vector x_n ; \mathbf{s} es el vector y_n ; y \mathbf{a} es el vector z_n , por lo que los elementos de la trama herramienta quedan:

$${}^0T_6 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Queremos averiguar la orientación náutica Z-Y'-X'' referida a S_0 tratando al vector \mathbf{a} como homólogo a \mathbf{x}_0 , y al vector \mathbf{n} como homólogo a \mathbf{z}_0 .

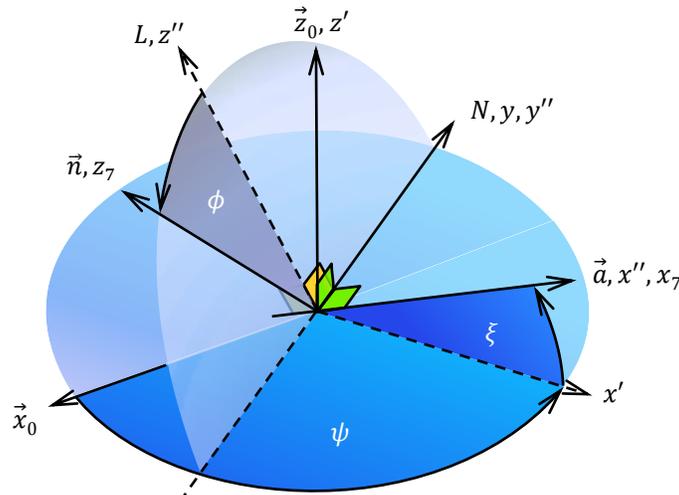


Figura 2.14. Orientación del elemento terminal

Observando el diagrama 2.14, se obtienen las siguientes relaciones con el producto punto de vectores:

$$\cos \psi = \frac{(\mathbf{x}' \cdot \mathbf{x}_0)}{|\mathbf{x}'||\mathbf{x}_0|} = \frac{(\mathbf{y}_0 \cdot \mathbf{N})}{|\mathbf{y}_0||\mathbf{N}|}$$

$$\cos \xi = \frac{(\mathbf{x}' \cdot \mathbf{a})}{|\mathbf{x}'||\mathbf{a}|} = \frac{(\mathbf{z}_0 \cdot \mathbf{L})}{|\mathbf{z}_0||\mathbf{L}|}$$

$$\cos \phi = \frac{(\mathbf{L} \cdot \mathbf{n})}{|\mathbf{L}||\mathbf{n}|} = \frac{(\mathbf{s} \cdot \mathbf{N})}{|\mathbf{s}||\mathbf{N}|}$$

Necesitamos obtener \mathbf{N} , \mathbf{L} , \mathbf{x}' y sus normas, $|\mathbf{N}|$, $|\mathbf{L}|$, $|\mathbf{x}'|$. Procedemos de acuerdo al método descrito en 2.4, para determinar la línea de nodos \mathbf{N} :

$$\mathbf{N} = \mathbf{z}_0 \times \mathbf{a}$$

$$\mathbf{N} = \mathbf{z}_0 \times \mathbf{z}_6$$

$$\mathbf{N} = \mathbf{y}_0'$$

$$\mathbf{N} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ 0 & 0 & 1 \\ a_x & a_y & a_z \end{vmatrix}$$

$$\mathbf{N} = \begin{vmatrix} 0 & 1 \\ a_y & a_z \end{vmatrix} \mathbf{i} - \begin{vmatrix} 0 & 1 \\ a_x & a_z \end{vmatrix} \mathbf{j} + \begin{vmatrix} 0 & 0 \\ a_x & a_y \end{vmatrix} \mathbf{k}$$

$$\mathbf{N} = -a_y \mathbf{i} + a_x \mathbf{j}$$

La norma de \mathbf{N} es entonces:

$$|\mathbf{N}|^2 = a_y^2 + a_x^2$$

Se calcula \mathbf{L} , vector que es perpendicular al vector \mathbf{a} y a la línea de nodos \mathbf{N} :

$$\mathbf{L} = \mathbf{a} \times \mathbf{N}$$

$$\mathbf{L} = \mathbf{z}_6 \times \mathbf{N}$$

$$\mathbf{L} = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ a_x & a_y & a_z \\ -a_y & a_x & 0 \end{vmatrix}$$

$$\mathbf{L} = \begin{vmatrix} a_y & a_z \\ a_x & 0 \end{vmatrix} \mathbf{i} - \begin{vmatrix} a_x & a_z \\ -a_y & 0 \end{vmatrix} \mathbf{j} + \begin{vmatrix} a_x & a_y \\ -a_y & a_x \end{vmatrix} \mathbf{k}$$

$$\mathbf{L} = -a_x a_z \mathbf{i} - a_y a_z \mathbf{j} + (a_x a_x + a_y a_y) \mathbf{k}$$

Para normalizar \mathbf{L} , se obtiene su norma:

$$|\mathbf{L}|^2 = (-a_x a_z)^2 + (-a_y a_z)^2 + (a_x a_x + a_y a_y)^2$$

$$|\mathbf{L}|^2 = a_x^2 a_z^2 + a_y^2 a_z^2 + (|\mathbf{N}|^2)^2$$

$$|\mathbf{L}|^2 = a_x^2 a_z^2 + a_y^2 a_z^2 + |\mathbf{N}|^2 |\mathbf{N}|^2$$

$$|\mathbf{L}|^2 = (a_y^2 + a_x^2) a_z^2 + |\mathbf{N}|^2 |\mathbf{N}|^2$$

$$|\mathbf{L}|^2 = a_z^2 |\mathbf{N}|^2 + |\mathbf{N}|^2 |\mathbf{N}|^2$$

$$|\mathbf{L}|^2 = (a_z^2 + |\mathbf{N}|^2) |\mathbf{N}|^2$$

$$|\mathbf{L}|^2 = (a_z^2 + a_y^2 + a_x^2)|\mathbf{N}|^2$$

$$|\mathbf{L}|^2 = |\mathbf{a}|^2|\mathbf{N}|^2$$

Los vectores \mathbf{a} , \mathbf{s} , \mathbf{n} de la trama herramienta, son unitarios, esto es: $|\mathbf{a}| = 1$, $|\mathbf{s}| = 1$, $|\mathbf{n}| = 1$, entonces:

$$|\mathbf{L}|^2 = |\mathbf{N}|^2$$

Se calcula \mathbf{x}' :

$$\mathbf{x}' = \mathbf{N} \times \mathbf{z}_0$$

$$\mathbf{x}' = \begin{vmatrix} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ -a_y & a_x & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

$$\mathbf{x}' = \begin{vmatrix} a_x & 0 \\ 0 & 1 \end{vmatrix} \mathbf{i} - \begin{vmatrix} -a_y & 0 \\ 0 & 1 \end{vmatrix} \mathbf{j} + \begin{vmatrix} -a_y & a_x \\ 0 & 0 \end{vmatrix} \mathbf{k}$$

$$\mathbf{x}' = a_x \mathbf{i} + a_y \mathbf{j}$$

Cuya norma es:

$$|\mathbf{x}'|^2 = a_y^2 + a_x^2 = |\mathbf{N}|^2$$

El ángulo entre \mathbf{L} y \mathbf{n} es el roll ó alabeo (rotación sobre el eje longitudinal o de acercamiento); El sentido del ángulo está indicado por la dirección del eje \mathbf{a} . Si el eje \mathbf{n} , tiene componente positiva sobre la línea de nodos \mathbf{N} , el ángulo medido es negativo:

$$|\mathbf{L}||\mathbf{n}| \cos \phi = \mathbf{L} \cdot \mathbf{n}$$

$$\phi = \arccos\left(\frac{\mathbf{L} \cdot \mathbf{n}}{|\mathbf{L}||\mathbf{n}|}\right); \text{ Si } (\mathbf{n} \cdot \mathbf{N}) > 0, \phi * = -1;$$

$$\phi = \arccos\left\{\frac{[-a_x a_z \mathbf{i} - a_y a_z \mathbf{j} + (a_x a_x + a_y a_y) \mathbf{k}] \cdot \vec{n}}{|\mathbf{L}||\mathbf{n}|}\right\}$$

$$\phi = \arccos\left\{\frac{-a_x a_z n_x - a_y a_z n_y + (a_x^2 + a_y^2) n_z}{|\mathbf{L}||\mathbf{n}|}\right\}$$

El ángulo entre \mathbf{L} y \mathbf{z}_0 es el pitch ó cabeceo (rotación sobre el eje transversal); el sentido del ángulo está indicado por la dirección de la línea de nodos \mathbf{N} . Si el eje \mathbf{L} tiene componente negativa sobre \mathbf{x}' (sombra de \mathbf{a} sobre plano xy), el ángulo medido es negativo:

El pitch también es el ángulo entre \mathbf{x}' y \mathbf{a} . Si el eje \mathbf{a} tiene componente positiva sobre \mathbf{z}_0 , el ángulo medido es negativo.

$$|\mathbf{L}||\mathbf{z}_0| \cos \theta = \mathbf{L} \cdot \mathbf{z}_0$$

$$\xi = \arccos\left(\frac{\mathbf{L} \cdot \mathbf{z}_0}{|\mathbf{L}||\mathbf{z}_0|}\right); \text{ Si } (\mathbf{L} \cdot \mathbf{x}') < 0, \xi * = -1; \text{ ó}$$

$$\xi = \arccos\left(\frac{\mathbf{x}' \cdot \mathbf{a}}{|\mathbf{x}'||\mathbf{a}|}\right); \text{ Si } (\mathbf{a} \cdot \mathbf{z}_0) > 0, \xi * = -1;$$

$$\xi = \arccos \left\{ \frac{(a_x \mathbf{i} + a_y \mathbf{j}) \cdot \mathbf{a}}{|\mathbf{x}'| \cdot |\mathbf{a}|} \right\}$$

$$\xi = \arccos \left\{ \frac{a_x a_x + a_y a_y}{|\mathbf{x}'| \cdot |\mathbf{a}|} \right\}$$

$$\xi = \arccos \left\{ \frac{a_x^2 + a_y^2}{|\mathbf{x}'| \cdot |\mathbf{a}|} \right\}$$

El ángulo entre \mathbf{x}_0 y \mathbf{x}' es el yaw ó guiñada (rotación sobre el eje vertical); si \mathbf{x}' tiene componente negativa sobre \mathbf{y}_0 , el ángulo medido es negativo:

$$|\mathbf{x}_0| |\mathbf{x}'| \cos \theta = \mathbf{x}_0 \cdot \mathbf{x}'$$

$$\psi = \arccos \left(\frac{\mathbf{x}_0 \cdot \mathbf{x}'}{|\mathbf{x}_0| |\mathbf{x}'|} \right); \text{ Si } (\mathbf{x}' \cdot \mathbf{y}_0) < 0, \psi \neq -1; \text{ ó}$$

$$\psi = \arccos(\mathbf{y}_0 \cdot \mathbf{y}'); \text{ Si } (\mathbf{y}' \cdot \mathbf{x}_0) < 0, \psi \neq -1;$$

$$\psi = \arccos \left\{ \frac{\mathbf{i} \cdot a_x \mathbf{i}}{|\mathbf{x}_0| \cdot |\mathbf{x}'|} \right\}$$

$$\psi = \arccos \left\{ \frac{a_x}{\sqrt{a_x^2 + a_y^2}} \right\}$$

Finalmente (ver apéndice 6.1)

$$x = d_6(c_1 c_2 c_3 c_4 s_5 - c_1 s_2 s_3 c_4 s_5 - s_1 s_4 s_5 + c_1 c_2 s_3 c_5 + c_1 s_2 c_3 c_5) + a_3(c_1 c_2 c_3 - c_1 s_2 s_3) + a_2 c_1 c_2$$

$$y = d_6(s_1 c_2 c_3 c_4 s_5 - s_1 s_2 s_3 c_4 s_5 + c_1 s_4 s_5 + s_1 c_2 s_3 c_5 + s_1 s_2 c_3 c_5) + a_3(s_1 c_2 c_3 - s_1 s_2 s_3) + a_2 s_1 c_2$$

$$z = d_6(s_2 c_3 c_4 s_5 + c_2 s_3 c_4 s_5 + s_2 s_3 c_5 - c_2 c_3 c_5) + a_3(s_2 c_3 + c_2 s_3) + a_2 s_2$$

$$\psi = \arccos \left\{ \frac{a_x}{\sqrt{a_x^2 + a_y^2}} \right\}$$

$$\xi = \arccos \left\{ \frac{a_x^2 + a_y^2}{|\mathbf{x}'| |\mathbf{a}|} \right\}$$

$$\phi = \arccos \left\{ \frac{-a_x a_z n_x - a_y a_z n_y + (a_x^2 + a_y^2) n_z}{|\mathbf{L}| |\mathbf{n}|} \right\}$$

Dónde:

$$|\mathbf{L}| = \sqrt{(a_x a_z)^2 + (a_y a_z)^2 + (a_x a_x + a_y a_y)^2}$$

$$|\mathbf{x}'| = \sqrt{a_x^2 + a_y^2}$$

$$|\mathbf{a}| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

$$\begin{aligned}
|\mathbf{n}| &= \sqrt{n_x^2 + n_y^2 + n_z^2} \\
a_x &= c_1 c_2 c_3 c_4 s_5 + c_1 c_2 s_3 c_5 - c_1 s_2 s_3 s_5 + c_1 s_2 c_3 c_5 - s_1 s_5 \\
a_y &= s_1 c_2 c_3 c_4 s_5 + s_1 c_2 s_3 c_5 - s_1 s_2 s_3 s_5 + s_1 s_2 c_3 c_5 + c_1 s_5 \\
a_z &= s_1 c_3 s_5 + s_1 s_3 c_5 + c_2 s_3 s_5 - c_2 c_3 c_5 \\
n_x &= c_1 c_2 c_3 c_4 c_5 c_6 - c_1 s_2 s_3 c_4 c_5 c_6 + c_1 c_2 c_3 s_4 s_6 - c_1 s_2 s_3 s_4 s_6 \\
&\quad - c_1 c_2 s_3 s_5 c_6 - c_1 s_2 c_3 s_5 c_6 - s_1 s_4 c_5 c_6 + s_1 c_4 s_6 \\
n_y &= s_1 c_2 c_3 c_4 c_5 c_6 - s_1 s_2 s_3 c_4 c_5 c_6 + s_1 c_2 c_3 s_4 s_6 - s_1 s_2 s_3 s_4 s_6 \\
&\quad - s_1 c_2 s_3 s_5 c_6 - s_1 s_2 c_3 s_5 c_6 + c_1 s_4 c_5 c_6 - c_1 c_4 s_6 \\
n_z &= s_1 c_3 c_4 c_5 c_6 + c_2 s_3 c_4 c_5 c_6 + s_1 c_3 s_4 s_6 - s_1 s_3 s_5 c_6 \\
&\quad + c_2 s_3 s_4 s_6 + c_2 c_3 s_5 c_6
\end{aligned}$$

2.6 Solución geométrica de cinemática Inversa de una cadena cinemática con seis grados de libertad

A continuación se muestra un método geométrico sintetizado de [4](Barrientos cap. 4), [10](Craig, cap. 4) y [33](Spong cap. 4) para obtener la solución en forma cerrada $q_k = f_k(x, y, z, \alpha, \beta, \gamma)$ para $k = 1 \dots 6$, de los parámetros articulares de una configuración robótica de seis grados de libertad. El método consiste en resolver por separado la cinemática inversa para posición y orientación. Esto se conoce como desacople cinemático [4](Barrientos p.118). Se considera que los primeros tres grados de libertad contribuyen a la posición, y los últimos 3 contribuyen a la orientación de la trama herramienta.

El desacople cinemático es posible si los ejes de revolución de las últimas tres juntas coinciden en un punto, situación descrita en [10](Craig, p.114). El modelo cinemático considerado posee éstas características.

Sea w , el punto correspondiente al origen de los sistemas S_3, S_4 y S_5 :

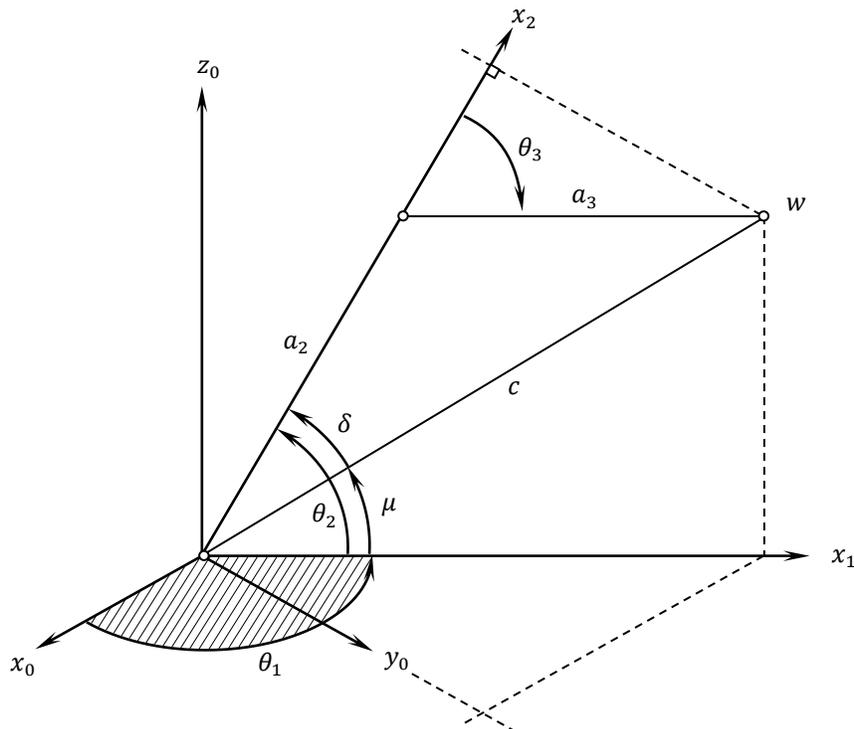


Figura 2.15 Obtención de los primeros tres GDL

Observando la figura 2.15 y por ley de cosenos:

$$c^2 = a_2^2 + a_3^2 - 2a_2a_3 \cos(\pi - \theta_3)$$

$$c^2 = a_2^2 + a_3^2 - 2a_2a_3(\cos \pi \cos \theta_3 + \sin \pi \sin \theta_3)$$

$$c^2 = a_2^2 + a_3^2 + 2a_2a_3 \cos \theta_3$$

$$2a_2a_3 \cos \theta_3 = c^2 - a_2^2 - a_3^2$$

$$\cos \theta_3 = \frac{c^2 - a_2^2 - a_3^2}{2a_2a_3}$$

$$\theta_3 = \arccos\left(\frac{c^2 - a_2^2 - a_3^2}{2a_2a_3}\right)$$

De la figura 2.15:

$$\delta = \arctan\left(\frac{a_3 \sin \theta_3}{a_3 \cos \theta_3 + a_2}\right)$$

$$\mu = \arctan\left(\frac{w_z}{\sqrt{w_x^2 + w_y^2}}\right)$$

Entonces:

$$\theta_2 = \delta + \mu$$

Nuevamente, de la Figura 2.15

$$\theta_1 = \arctan\left(\frac{w_y}{w_x}\right)$$

Las coordenadas (w_x, w_y, w_z) de w se determinan a partir de la posición (o_x, o_y, o_z) y orientación náutica Z-Y'-X'' (ψ, ξ, ϕ) de alguna trama objeto $S_7: \{x_7, y_7, z_7\}$.

Sea (x', y', z') un marco de referencia paralelo a S_0 . Si S_7 es alcanzable, entonces $\mathbf{P} = (P_x, P_y, P_z) = (o_x, o_y, o_z)$.

A partir de la figura 2.6.2:

$$w_x = P_x - d_6 \cos \xi \cos \psi$$

$$w_y = P_y - d_6 \cos \xi \sin \psi$$

$$w_z = P_z - d_6 \sin \xi$$

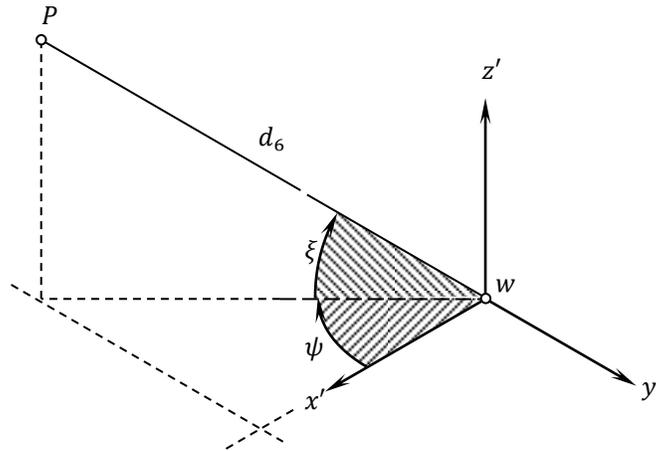


Figura 2.16 Ubicación de w a partir de P

Durante la operación del mecanismo prototipo se considera $\psi = 0$, pues carece de éste grado de libertad. Ocurren singularidades cuando $c > a_2 + a_3$. Esto se le conoce como desigualdad de triángulo. Sucede cuando la combinación de P con alguna orientación (náutica) Tait-Bryan Z-Y'-X'' (ψ, ξ, ϕ) provocan la ubicación del origen de la muñeca w fuera del alcance de los primeros tres grados de libertad. Sin embargo

existe un punto w' dentro del espacio de trabajo, que es el más cercano a w (Para una solución detallada ver sección 2.7).

Si se observan las ecuaciones para θ_1 y μ , también ocurren singularidades cuando $w_x = w_y = 0$.

Para la solución de los últimos tres grados de libertad se parte del sistema S_3 ubicado en w . Utilizando el método delineado en la sección 2.4, obtenemos la orientación náutica Z-Y'-X'' del sistema S_6 ubicado en el punto de operación P , referida a S_3 . El vector de acercamiento \vec{a} de la trama herramienta S_6 con matriz homogénea 0T_6 corresponde al vector z_6 DH en el sistema S_6 .

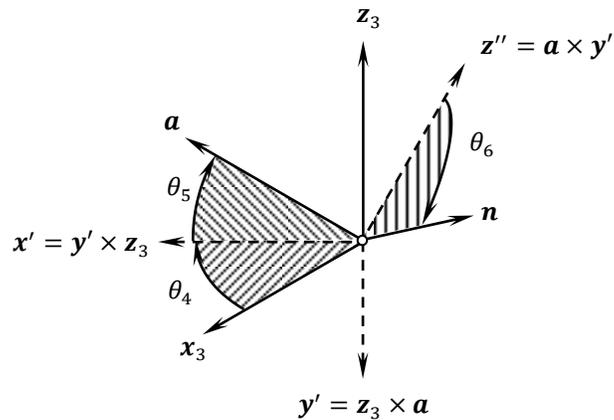


Figura 2.17 Obtención de los últimos tres grados de libertad

$$\theta_4 = \arccos(x_3 \cdot x');$$

Si $(x' \cdot y_3) < 0$, Cambiar el signo de θ_4

$$\theta_5 = \arccos(z'' \cdot z_3);$$

Si $(z'' \cdot x') < 0$, Cambiar el signo de θ_5

$$\theta_6 = \arccos(z'' \cdot n);$$

Si $(n \cdot y') > 0$, Cambiar el signo de θ_6

2.7 Espacio de Trabajo Restringido

El espacio de trabajo de un manipulador son los puntos en \mathbb{R}^3 del espacio tarea, que son alcanzables con el efector final. El espacio de trabajo diestro es el subespacio del espacio de trabajo en el que se pueden alcanzar todas las posiciones con orientación arbitraria. Debido a limitaciones mecánicas y geométricas en los actuadores y geometría de los eslabones, no todas las posiciones del espacio de trabajo pueden ser alcanzadas con orientación arbitraria. Este es el espacio de trabajo restringido, el complemento del espacio de trabajo diestro. [1](Arking (Lynch, Kantor) p.40).

El conjunto de posibles orientaciones y posturas forma el espacio de configuración y éste es función de la posición del punto de operación P

El método utilizado en la solución por cinemática inversa de los seis parámetros articulares de la sección 2.5, no considera el caso en el que w , punto donde convergen los orígenes de los sistemas S_3, S_4, S_5 , no se encuentra dentro del espacio de trabajo. El problema consiste en encontrar un punto w' dentro del espacio de trabajo que sea el más cercano a w . Existen múltiples selecciones de w' , algunas de ellas discontinuas

En la Figura 2.18 se observa uno de los primeros intentos por visualizar el espacio restringido

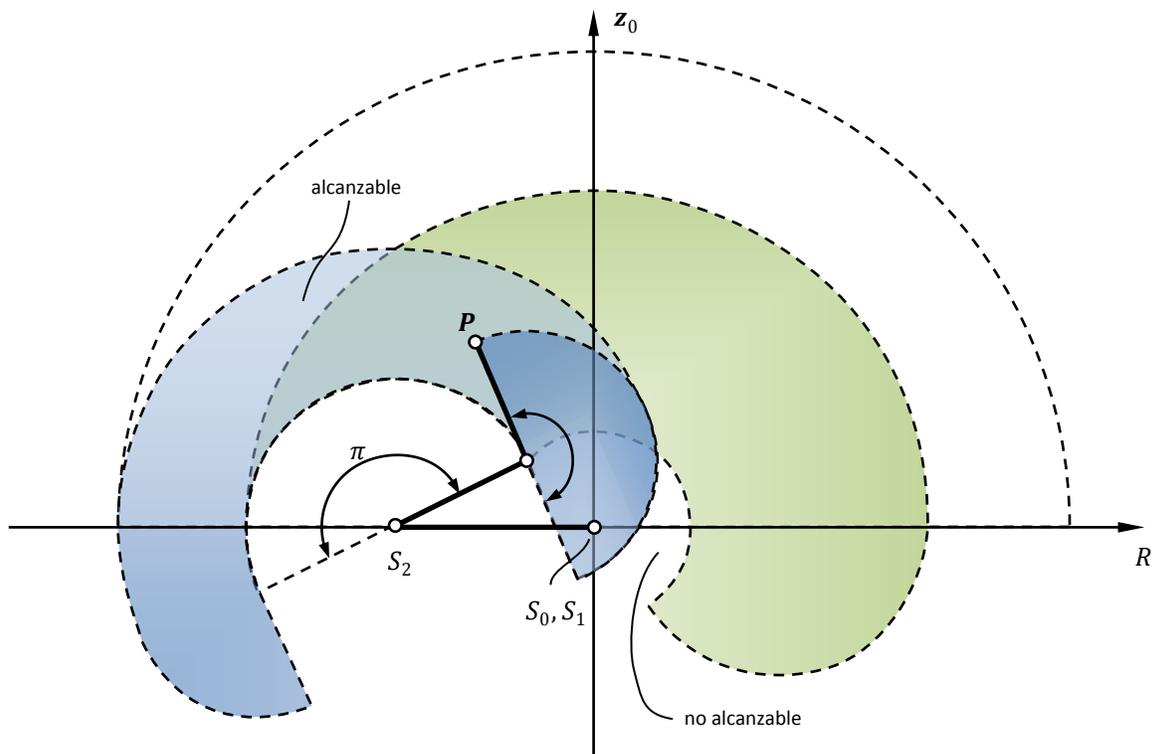


Figura 2.18. Boceto del espacio de trabajo

La figura 2.19 muestra el cono de ángulo máximo permitido por un espacio de trabajo de seis grados de libertad, para la configuración robótica analizada
 Defínase $S_7: \{x_7, y_7, z_7\}$, como la trama que se desea alcanzar

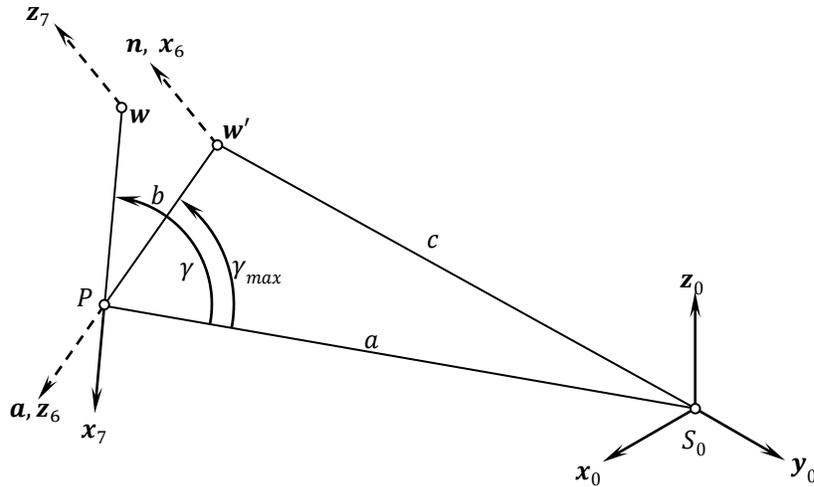


Figura 2.19. Cono de ángulo máximo

- P : punto de operación
- a : longitud entre el punto de operación y el origen de la base del robot S_0
- b : longitud de la muñeca (d_6 DH)
- c : suma de las longitudes del brazo y el antebrazo (primer y segundo eslabón; a_2 a_3 DH) equivale al límite del alcance de los primeros tres grados de libertad, cuyo espacio de trabajo esférico tiene un radio de c
- γ_{max} : ángulo máximo
- γ : ángulo entre \overline{Pw} y $\overline{PS_0}$
- x_7 : coordenada homóloga a " x_0 " para el sistema de coordenadas con orientación náutica arbitraria
- \mathbf{a} : vector de acercamiento de la trama herramienta 0T_6
- w : coordenadas de la muñeca desde las cuales el punto de operación tendría la orientación solicitada
- w' : coordenadas de la muñeca corregidas para estar dentro del espacio de trabajo; es el punto desde el cual el punto de operación tiene la orientación más cercana a la solicitada

Por ley de cosenos:

$$c^2 = a^2 + b^2 - 2ab \cos(\gamma_{max})$$

$$\gamma_{max} = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

Existen puntos dentro del espacio de trabajo que no pueden ser alcanzados con una orientación arbitraria. La combinación de guiñada (yaw) y cabeceo (pitch) da un ángulo γ que depende del punto de operación. Sea P el punto de operación, entonces γ es el ángulo que forma el segmento $\overline{PS_0}$ con el segmento \overline{Pw} . Si γ es mayor que un valor γ_{max} que depende de la ubicación del P dentro del espacio de trabajo, P es inalcanzable con esa orientación.

Se puede observar que las soluciones alcanzables, se encuentran en la intersección de dos conos cuyas generatrices son los lados c y b y que tienen el lado a como eje de rotación. del triángulo formado por el punto de operación, el origen del sistema raíz S_0 , y el conjunto de puntos w' como eje de rotación

Diremos que S_7 , es inalcanzable si el conjunto ó la combinación de posición y orientación no tiene solución dentro del espacio de trabajo. Si la orientación de S_7 , es inalcanzable, pero P está dentro del espacio de trabajo, existe una orientación que es la más cercana a la inalcanzable. La corrección se realiza a partir de la diferencia entre el valor γ y γ_{max}

Se determina la diferencia de ángulo $\gamma_{max} - \gamma$, y se rota el vector \overrightarrow{Pw} alrededor de la normal a $\overline{PS_0}$ y \overline{Pw} en el sentido de la diferencia. Se efectúa la suma vectorial $P + \overline{w}$. La resultante es w' .

Sólo queda deducir la nueva orientación náutica de S_7 , a partir de w'

Los pasos son los siguientes. Utilizando la nomenclatura de la sección 2.4, denotemos por \mathbf{a} al vector de acercamiento \mathbf{z}_6 del marco de referencia S_6 ubicado en P y fijo en la herramienta terminal. Entonces $\mathbf{a} = \mathbf{z}_6 = P - w'$. Denotemos por \mathbf{n} al vector perpendicular al vector de acercamiento \mathbf{a} y al vector deslizante \mathbf{s} del marco herramienta S_6 , es decir $\mathbf{n} = \mathbf{a} \times \mathbf{s} = \mathbf{x}_6$. Falta determinar $\mathbf{n} = \mathbf{x}_6$, componente homóloga a \mathbf{z}_0 para el sistema $S_6: \{x_6, y_6, z_6\}$

Como recordatorio, el sistema $S_7: \{x_7, y_7, z_7\}$ proviene de una orientación náutica arbitraria Z-Y'-X'' referida al sistema S_0 con valores de yaw, pitch y roll denotados por el vector simbólico (ψ, θ, ϕ) . El sistema $\{x_7, y_7, z_7\}$ tiene entonces una orientación de (ψ, θ, ϕ) que es inalcanzable. Ésta última orientación está referida al sistema S_0 , por lo que se denotará como ZYX_07. Se quiere encontrar una orientación alcanzable (ψ', θ', ϕ') , referida a S_0 , que se denotará como ZYX_06, más cercana a ZYX_07 (ψ, θ, ϕ) , de modo que P se encuentre dentro del espacio de trabajo (sea alcanzable). Se define como $S_6: \{x_6, y_6, z_6\}$ al sistema que tiene orientación (ψ', θ', ϕ') , El vector que se busca, $\mathbf{z}_6 = \mathbf{a}$, tiene la dirección del vector $P - w'$.

Nos es de interés la determinación de $\mathbf{n} = \mathbf{x}_6$. Existen varias soluciones, se eligió por conveniencia la dirección de la componente perpendicular de \mathbf{z}_7 sobre \mathbf{a} . Aunque también podría optarse por rotar \mathbf{z}_7 alrededor de la normal a $\overline{PS_0}$ y \overline{Pw} en el sentido $\gamma_{max} - \gamma$, del mismo modo que se giró el vector \overrightarrow{Pw} para encontrar w' .

Con \mathbf{z}_6 y \mathbf{x}_6 se completa el nuevo sistema $S_6: \{x_6, y_6, z_6\}$ que tiene por construcción una orientación alcanzable y que es la más cercana al marco con orientación arbitraria S_7 . A partir de S_6 siguiendo el procedimiento de la sección 2.2 para orientación y 2.4 para cinemática inversa, se determinan los parámetros articulares \mathbf{q} .

En la implementación, el algoritmo determina y realiza una comparación entre γ y γ_{max} y cambia el flujo del programa de acuerdo al resultado.

En la Figura 2.20 se muestra el ángulo máximo permitido por un espacio de trabajo de cinco GDL. Éste ángulo máximo depende de la posición P

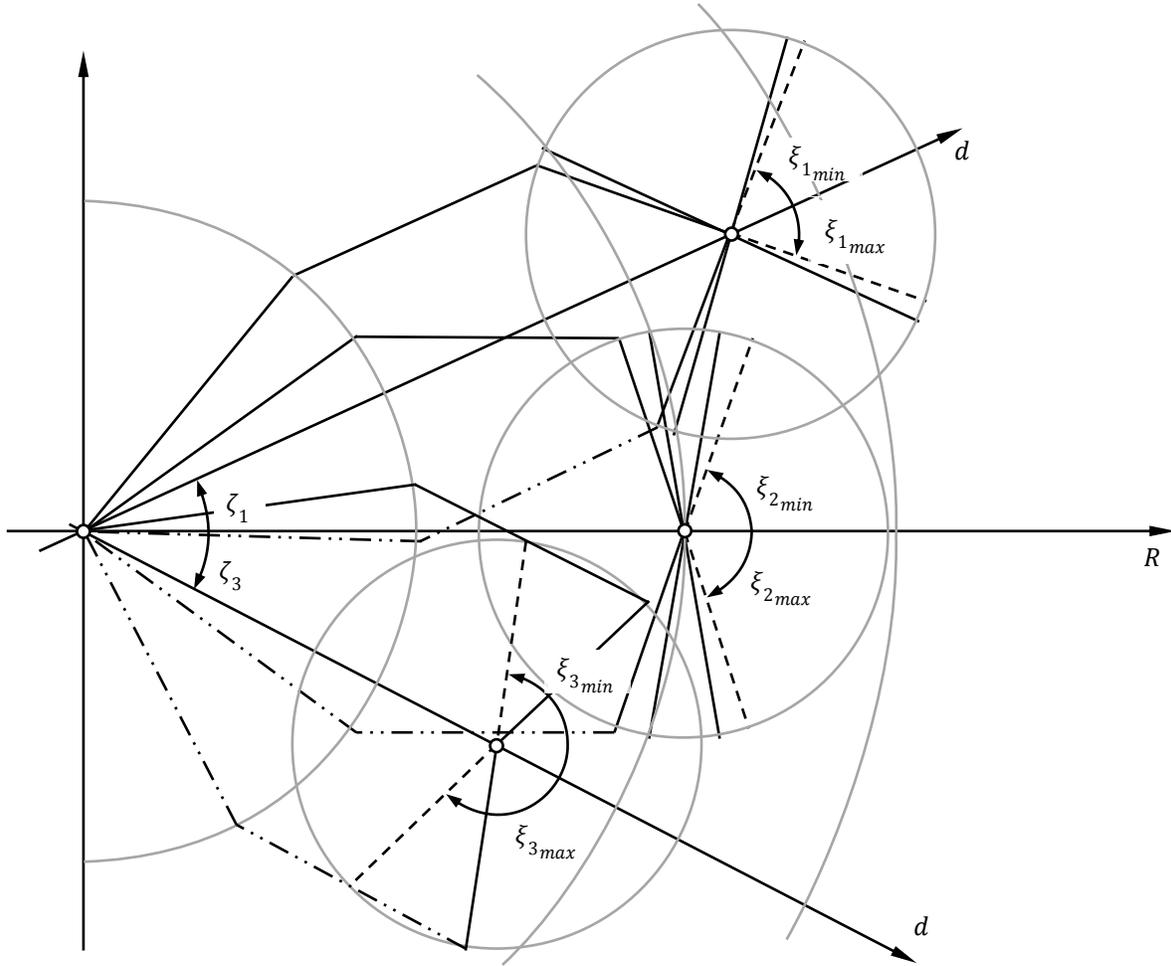


Figura 2.20 Variación del pitch máximo ξ_{max} para distintas posiciones P

2.8 Control cinemático

Una vez obtenida una solución para la cinemática inversa, se puede proceder con el control cinemático. El control cinemático consiste en la generación de trayectorias (curvas; caminos; conjunto continuo de puntos, que deben ser recorridos a velocidades y en intervalos de tiempo específicos) que debe seguir cada articulación del robot, que se traducen en movimientos y posturas específicas del elemento terminal (esto es, en espacio tarea), para cumplir con los objetivos de alguna tarea, como se menciona en [4](Barrientos p.169). Se deben tomar en cuenta las restricciones mecánicas de las articulaciones impuestas por los actuadores, dimensiones de los eslabones, es decir el espacio de trabajo restringido, durante la generación de estas trayectorias. Las trayectorias generadas deben pertenecer al espacio de trabajo restringido. El control cinemático está intrínsecamente relacionado con planeación de movimiento y navegación, por ejemplo, evasión de obstáculos y colisiones, como se discute en [1](Arkin p.39)

2.9 Acoplamiento virtual de objetos

Acoplamiento o anclaje de objetos es un algoritmo para manipular computacionalmente la descripción cinemática de un objeto virtual.

Se entiende por acoplamiento a una relación mecánica (restricción geométrica) que permanece constante entre dos objetos. De esta forma el objeto acoplado queda restringido en su movimiento al reducir sus grados de libertad, heredando las transformaciones mecánicas sufridas por un objeto acoplante y añadiendo nuevas restricciones a la cinemática del objeto acoplante, es decir, reduciendo su espacio de trabajo.

Consiste en lo siguiente:

Sea P el origen de un sistema de coordenadas G asociado a un cuerpo rígido en el espacio.

P sufre transformaciones mecánicas (cinemáticas o de cuerpo rígido, ya que no se está tomando en cuenta fuerza ni masa): traslación y rotación, por lo que a P se le asocia una matriz M

Sea A una matriz homogénea que representa un sistema A ortonormal derecho arbitrario ubicado en un punto Q ; G es la matriz homogénea que representa la transformación actual aplicada a un vector constante P_0 . P es el vector resultante de aplicar la transformación G a P_0 . P es una versión rotada y trasladada de P_0 : P descrito desde el sistema G , es indistinto de P_0 . Esta rotación y traslación están codificadas en G (de acuerdo al procedimiento explicado en párrafos anteriores). Intuitivamente, el sistema G presenta una orientación y localización distinta para el sistema A ubicado en Q . Se denomina X a la matriz homogénea que codifica la orientación y localización de G respecto de A . Se quiere encontrar una transformación X , que relacione A con G . El planteamiento matemático de lo anterior se expresa como: A transformado por X es igual a G . Ó: para alinear A con G es necesaria la transformación geométrica X . La ecuación matricial a resolver es la siguiente:

$$XA = G$$

Despejando la transformación incógnita X :

$$X = GA^{-1}$$

Para comprobar que efectivamente X transforma A en G , se realiza la sustitución:

$$(GA^{-1})A = G$$

$$GA^{-1}A = G$$

$$GI = G$$

$$G = G$$

X representa una relación espacial entre A y G . X es entonces una transformación rígida; constante. Esto es, una vez determinada no cambia. Si ahora se utiliza X para representar un anclaje con un sistema G , se deben observar varias cosas. X es única y constante durante todo el tiempo del anclaje, y A es una matriz variable.

Conforme A varía, X permanece constante. Si X se utiliza para localizar un objeto en relación con A , variaciones en A se reflejan en el objeto. Al momento de desanclar el objeto de A , la matriz G asociada al objeto tiene que ser igual a X multiplicado por el último valor de A .

Entonces, utilizando la nomenclatura anterior, se tiene en el espacio un objeto (virtual o su representación) con localización y orientación representados por G . Si se desea interactuar con el objeto; manipularlo mecánicamente, se hace por medio de una matriz variable A . Al momento del anclaje o sujeción del objeto, se determina un ancla $X = G(A_{inicial})^{-1}$ (tiene que verificarse que el objeto esté dentro del alcance o espacio de trabajo del sistema A). En ese instante, se sustituye la matriz G asociada al objeto por XA . Si A no varía, el objeto permanece en la misma posición ya que el ancla X es única y constante para cada anclaje. Si entonces A varía, el objeto hereda las transformaciones sufridas por A , y cambian su posición y orientación. Si A representa a una cadena cinemática se debe considerar que en este estado ciertas configuraciones pueden ser inalcanzables debido a las dimensiones del objeto, debido a otros obstáculos presentes, restricciones cinemáticas, cercanía con la frontera del espacio de trabajo, cercanía con puntos singulares, etc. Una vez que se ha manipulado al objeto, éste ha alcanzado una posición y orientación XA_{final} . El desanclaje se consuma guardando este valor en la matriz G asociada al objeto, para comenzar un nuevo ciclo.

Se puede refinar y extender este algoritmo considerando X como variable dentro de un rango, y aplicando restricciones diferenciales como inercia lineal, inercia rotacional, fricción, impedancia, a las trayectorias posibles de A y al rango de X .

En programación un API (Application Programming Interface) es una especificación estándar de comunicación con algún dispositivo electrónico, o algún componente de software como una biblioteca o un servicio de Internet, o los servicios proveídos por un sistema operativo.

En el caso de programación de una aplicación tridimensional, se facilita el desarrollo y se acelera la ejecución mediante el uso de una tarjeta de vídeo, esto en general requiere del uso de un API de gráficos. Los dos más importantes son: "OpenGL" y "DirectX". DirectX es un API propiedad de Microsoft y es exclusivo del sistema operativo Windows, mientras que OpenGL es multiplataforma y de libre acceso.

Existen diversas alternativas en cuanto a lenguajes e infraestructuras de software (SDK, Software development kit) sobre las cuales se puede construir algún tipo de interfaz desde donde se pueda controlar un dispositivo mecatrónico como el caso del robot prototipo

C# pertenece a

La principal ventaja de C# sobre C++ es la administración de recursos de memoria. En C++ se debe invertir tiempo de desarrollo escribiendo y corrigiendo (depurando) clases que administren la memoria. Otra de las ventajas de C# sobre C++, es que en C++ se debe de implementar funcionalidad "estructural" para brindar soporte a operaciones muy comunes, como guardar y cargar archivos, conexión a un puerto serial, pero no disponibles de forma estándar. En C++ Se tiene que empezar completamente desde cero, aunque existen librerías de terceros que ya proveen esta funcionalidad, pero que se requiere de la documentación disponible. El uso de C++ está más orientado a proyectos profesionales, dada su dificultad y tiempo de desarrollo. Se utiliza generalmente para optimizar partes críticas de una aplicación.

Matlab por su parte, cuenta con utilidades denominadas cajas de herramientas que extienden la funcionalidad, aunque se depende de la versión de Matlab instalada en el equipo.

Las alternativas de desarrollo de DirectX en C# son XNA4, SharpDX, SlimDX.

Se optó por utilizar el IDE (Integrated Design Environment) "Visual Studio C# express 2010" con NET Framework 4.0, en conjunto con la biblioteca XNA 4.0

Algunas bondades de C# son: acceso al API completo para plataformas Windows, diversas funciones de comunicación serial, comunicación por internet, captura de vídeo, guardar y cargar archivos, etc.

El método para representar objetos tridimensionales en la computadora y animarlos, matemáticamente y computacionalmente es equivalente al método utilizado en la resolución de cinemática directa e inversa. Se utilizan matrices homogéneas y ángulos de Euler para representar posturas de objetos, con un paso adicional de proyección de la escena virtual a un plano de cámara, llevado a cabo también con una matriz homogénea. (detalles en [Lengyel] y [Riemer]).

A cada objeto virtual se le asocia una matriz homogénea que representa su situación cinemática respecto de un marco de referencia arbitrario dentro del entorno virtual. Sobre este marco de referencia se describen la postura y orientación de todos los objetos virtuales, que se codifica dentro de la matriz de cada objeto

Toda interacción de objetos dentro del ambiente virtual, como cambio de posición y orientación, se reflejan dentro de los valores numéricos de las matrices asociadas a cada objeto. De esta forma, se consigue la simulación de colisiones, inercia, gravedad, etc.

La visualización de este entorno virtual, corresponde a la proyección tridimensional de la geometría modelada de todos los objetos representados a un cuadro o marco. El punto de vista (posición y orientación) de este cuadro, también queda codificado por una matriz homogénea.

Los pasos a seguir son: la transformación en el espacio de los vértices del modelo, por medio de las matrices de postura. Posteriormente transformar los vértices al espacio cámara, multiplicando por la matriz homogénea asociada a la cámara, y finalmente multiplicar por la matriz de proyección

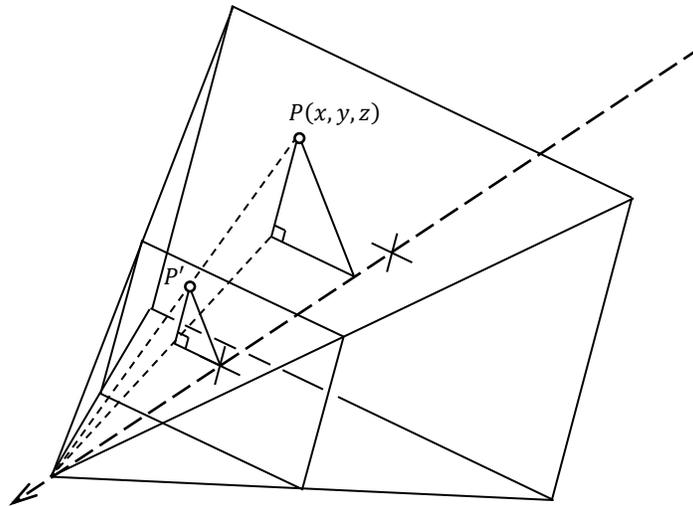


Figura 3.1. Proyección de P en P'

El método para obtener la matriz de proyección consiste en lo siguiente:

Se supone que la escena virtual se encuentra contenida en un poliedro piramidal, como se muestra en la figura 3.1, también llamado tronco, limitado por 6 planos: frontal, fondo, superior, inferior, derecho, e izquierdo, denotados por n, f, t, b, r, l , respectivamente. Todos los puntos P dentro del poliedro se describen relativos a un marco de referencia derecho ubicado en el vértice de la pirámide. La postura del tronco queda codificado por una matriz homogénea V .

Se desea obtener un mapeo de la pirámide a un prisma rectangular con rango $-1 < x' < 1, -1 < y' < 1, -1 < z' < 1$

Las coordenadas de la proyección de P sobre el plano cercano ubicado en $z = -n$, se obtienen por semejanza de triángulos:

$$x = -\frac{n}{P_z} P_x$$

$$y = -\frac{n}{P_z} P_y$$

Las coordenadas (x, y) están acotadas por r, l, t, b , por lo que satisfacen $l \leq x \leq r; b \leq y \leq t$. La transformación de (x, y) en (x', y') es lineal:

$$x' = \left(\frac{x-l}{r-l} \right) 2 - 1$$

$$y' = \left(\frac{y-b}{t-b} \right) 2 - 1$$

Sustituyendo y simplificando:

$$x' = \left(\frac{-2n}{r-l} \right) \frac{P_x}{P_z} - \frac{r+l}{r-l}$$

$$y' = \left(\frac{-2n}{t-b} \right) \frac{P_y}{P_z} - \frac{t+b}{t-b}$$

Por un procedimiento similar se obtiene z' :

$$z' = \left(\frac{2nf}{f-n} \right) \frac{1}{P_z} - \frac{f+n}{f-n}$$

Agrupando las transformaciones anteriores en una matriz resulta: la matriz de proyección H :

$$H = \begin{pmatrix} -\frac{2n}{r-l} & -\frac{l+r}{l-r} & 0 \\ 0 & -\frac{2n}{t-b} - \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} - \frac{2nf}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

El lector interesado puede consultar [19](Lengyel pp. 121-125) para una discusión comprensiva acerca de la matriz de proyección y gráficos por computadora.

Experimentación y Resultados

4.1 Comprobación numérica de las fórmulas obtenidas de cinemática directa

Se puede verificar que las fórmulas obtenidas en 2.3 son correctas. La postura de un marco de referencia $\{a^*, s^*, n^*\}$ obtenido a partir del cómputo de los parámetros de la fórmula coincide con la postura de un marco de referencia $\{a, s, n\}$ obtenido por medio de multiplicación matricial numérica de las matrices de postura individuales de cada eslabón.

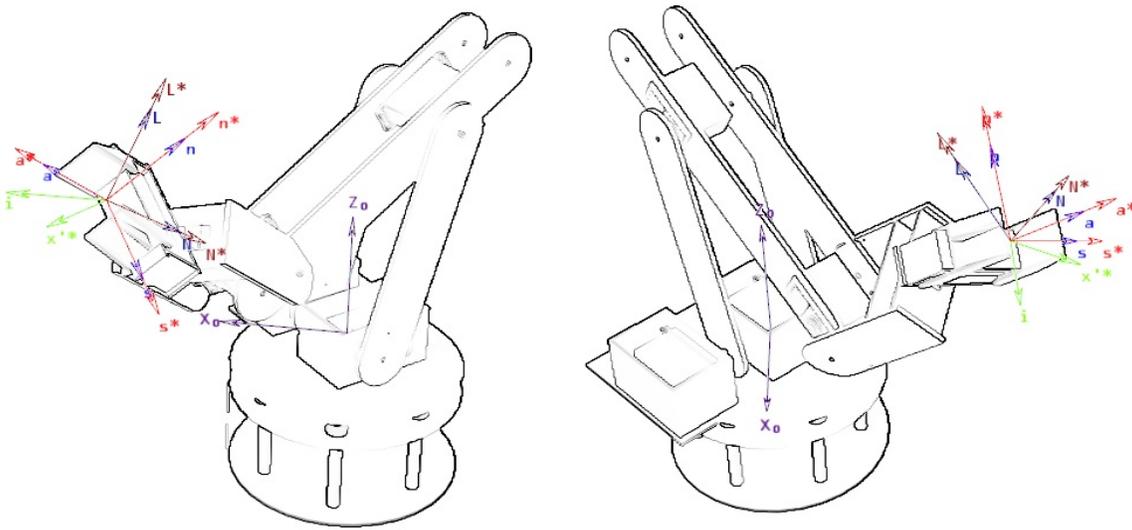


Figura 4.1 Captura de la plataforma virtual para dos vistas de la misma configuración

La figura 4.1 es una captura mejorada de la plataforma virtual. Los vectores: a^* , n^* , s^* , x^{**} , N^* , L^* y los vectores a , n , s , N , L mostrados, son los obtenidos por medio de las fórmulas en 2.3 y 6.1, y por medio de multiplicación matricial numérica respectivamente. Como puede observarse, a coincide con a^* , n coincide con n^* , etc. Los valores de postura del elemento terminal para la figura mostrada fueron $(x, y, z, \psi, \xi, \phi) = (124[mm], 289[mm], 115[mm], 53^\circ, -39^\circ, -33^\circ)$, para una configuración $\mathbf{q} = (q_1, q_2, q_3, q_4, q_5, q_6) = (41^\circ, 108^\circ, -140^\circ, 36^\circ, 159^\circ, -57^\circ)$, con longitudes de brazo antebrazo y muñeca de $136.3[mm]$, $135.7[mm]$, $128.2[mm]$, correspondientes con los parámetros denavit-hartenberg: a_2 , a_3 , d_6 respectivamente.

4.2 Actualización humana de ambiente virtual.

El experimento consiste en evadir obstáculos cuya actualización en el ambiente virtual se realiza manualmente. La representación virtual de la región tridimensional a evitar es una esfera de radio conocido. Durante la evasión, es necesaria la solución continua de cinemática inversa, solución que se codifica y se envía por puerto serial al microcontrolador, y que éste transforma en una señal de control que se envía a los servos. Se pone a prueba también, el funcionamiento en conjunto de la comunicación serial con el prototipo robótico.

El experimento se realizó con el propósito de demostrar la operación semiautomática durante la evasión de obstáculos. El operador traza una trayectoria cualquiera en el espacio de trabajo, y la computadora resuelve la determinación de los valores angulares de las articulaciones al mismo tiempo que evita la región esférica. Este tipo de control semiautomático puede encontrar aplicación en cirugía robótica, como las aplicaciones que se mencionan en el inciso 9 de la Tabla 1 de la sección 1.18. Aunque en los artículos donde vienen referidas las aplicaciones, no se mencionan este tipo restricciones virtuales.

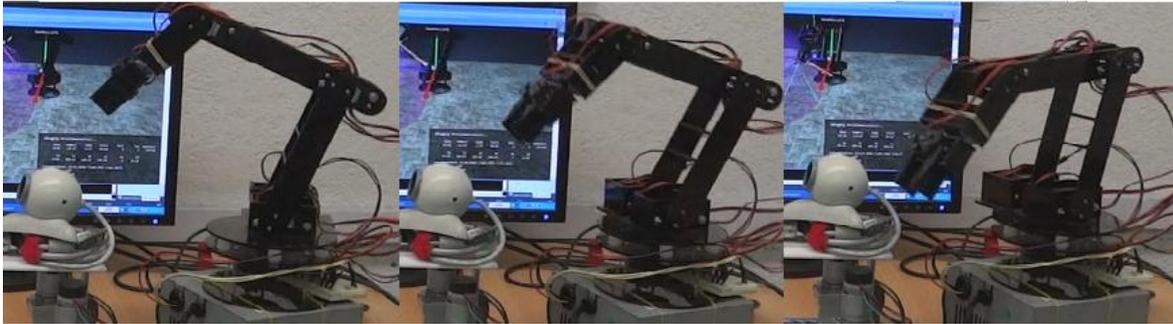


Figura 4.2 Evasión semiautomática de un obstáculo

En la figura 4.2 se observa la ejecución de una trayectoria cualquiera por un operador. Cuando algún punto de la trayectoria se encuentra dentro de la región esférica a evitar, se escoge arbitrariamente un punto en la frontera esférica y se utiliza éste en lugar del anterior. La secuencia de imágenes corresponde con la evasión de una “esfera fantasma” cuyo centro se actualizó manualmente en un entorno virtual, de manera que coincidiera con la esfera mostrada. El operador realiza movimientos que específicamente atraviesan la región a evadir, y el algoritmo corrige automáticamente la configuración del robot.

Se puede mejorar el algoritmo si en lugar de actualizar manualmente la posición del obstáculo, esta posición se obtiene mediante el uso conjunto de sensores inerciales inalámbricos, emisores infrarrojos y el procesamiento automático de video (visión por computadora).

La interfaz programada es capaz de establecer comunicación bidireccional con el robot prototipo y emitir comandos de control cinemático abierto. Tiene la capacidad de recibir información adicional de sensores como acelerómetros o unidades inerciales, por ejemplo el MinIMU de www.pololu.com, o de un mecanismo de captura, a través de un puerto serial secundario.

Su uso está en poder conocer y visualizar el estado a priori de la configuración cinemática utilizada para coordinar los movimientos reales de los eslabones del robot prototipo, así como resolver colisiones.

La cinemática inversa permite la manipulación virtual del Punto de Operación en espacio tridimensional euclidiano, es decir, en espacio tarea, con seis grados de libertad: tres traslaciones y tres rotaciones. Los parámetros articulares se resuelven automáticamente, se codifican serialmente y se envían por puerto serial al dispositivo remoto. Sobre el algoritmo de cinemática inversa, se construyen distintas restricciones al movimiento que facilitan la manipulación, no necesariamente presentes en el prototipo, pero si relacionadas con alguna tarea, por ejemplo, mantener el punto de operación siempre contenido en un plano paralelo a la dirección de la cámara virtual, o mantener el efector final fuera de una región, como en el caso de evasión de obstáculos.

La interfaz integra en el ambiente virtual información geométrica, cómo posición, ángulos, dimensiones de eslabones, orientación de Euler, información de la comunicación serial; con el propósito de monitorear y facilitar la corrección (depuración) de errores. El monitoreo del estado operacional es una de las funciones de control supervisado, como se mencionó en la introducción.

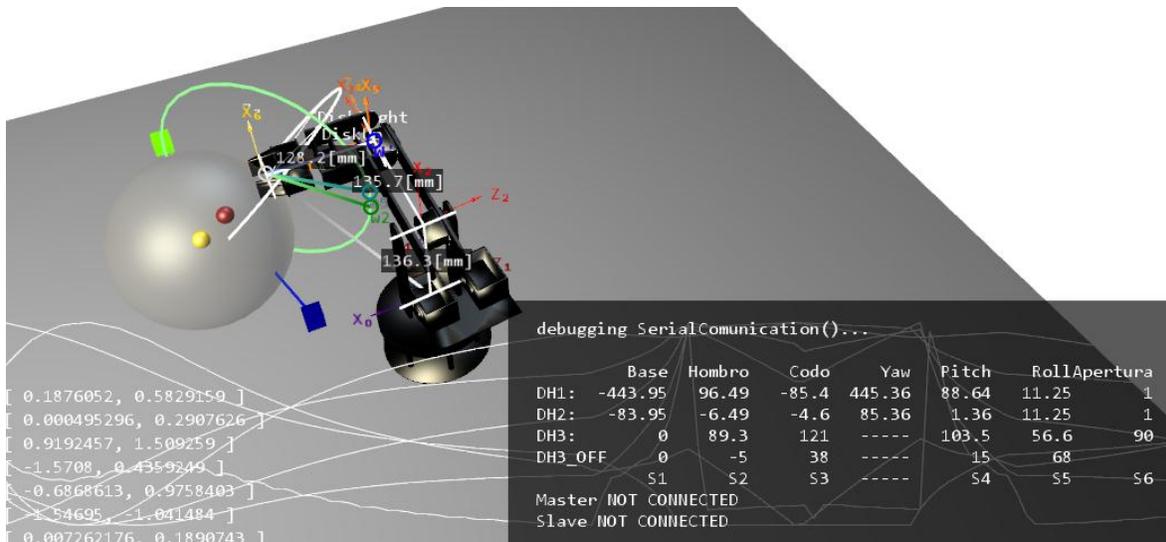


Figura 4.3. Actualización de la posición virtual del obstáculo, y evolución temporal del vector q

En la figura 4.3 se observan las gráficas de la evolución en el tiempo de los parámetros angulares q que son enviados por puerto serial al robot prototipo. Dado que los servos poseen distinto sentido de rotación y la configuración $\vec{q}_{servo} = \vec{0}$ tiene desfases, los parámetros articulares encontrados por cinemática inversa tienen que transformarse al espacio configuración del robot prototipo, que en el recuadro de la figura aparece denominado como DH3 para denotar denavit-hartenberg 3. Esta transformación es lineal, (estictamente una afinidad) y dado que también tiene traslación, se utiliza una matriz homogénea de siete dimensiones. El cambio de sentido en el eje de giro, se alcanza con un cambio de signo en los elementos diagonales que correspondería a un escalamiento.

Tanto el despliegue numérico como el gráfico sirven para corregir errores durante la operación. El más significativo de éstos, fue la discontinuidad paramétrica que sucede al considerar rangos de $0 < q_i \leq 2\pi$, que se corrigió considerando a los parámetros angulares de forma absoluta.

En la figura 4.3 también se muestran las distintas soluciones para cinemática inversa de cinco y seis grados de libertad, y la región esférica a evitar.

Con el propósito de asegurar una solución correcta de cinemática inversa de seis grados de libertad, se implementaron dos métodos y sus variantes: geométrico y matricial. El método algebraico consiste en despejar las variables de orientación y el vector de posición a partir de las ecuaciones del modelo cinemático directo. El método geométrico se discute en el capítulo dos, y el método matricial, desarrollado a continuación.

Se obtuvieron resultados mixtos con soluciones de forma parcial, y se optó por una combinación. Algunos métodos coincidían en los resultados y discrepaban en otros.

Entre los primeros intentos se presenta el matricial, que se discute en [4] y [33] sin embargo fue una solución parcial pues uno de los ángulos obtenidos discrepaba del obtenido por otros métodos:

Se partió del siguiente razonamiento:

La trama herramienta S_6 debe tener una orientación náutica respecto de S_0 . Es decir, debe de existir una secuencia de tres rotaciones intrínsecas consecutivas sobre los ejes de S_0 tales que lo alineen con S_6 .

S_6 también es el resultado de la cinemática directa.

Entonces, si se asume que la submatriz de rotación de la trama herramienta S_6 es el resultado de transformar al sistema S_0 primero por una rotación en z_0 de α radianes, luego por una rotación en y_0' de β radianes y finalmente por una rotación en x_0'' de γ radianes, el resultado será una matriz con los siguientes elementos.

Se puede entonces averiguar directamente la orientación Euler ZYX referido a la trama S_0 para cinemática directa utilizando esta ecuación. Una modificación sencilla del procedimiento se puede utilizar para deducir la orientación desde la trama S_3 cuyos ángulos se utilizan en la cinemática inversa. Se considera que el

vector se encuentra en representación renglón, por lo que se utiliza la transpuesta de las matrices de rotación y se postmultiplica:

$$R_z^T(\alpha)R_y^T(\beta)R_x^T(\gamma)$$

$$\begin{bmatrix} c_\alpha & -s_\alpha & 0 \\ s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} c_\beta & 0 & s_\beta \\ 0 & 1 & 0 \\ -s_\beta & 0 & c_\beta \end{bmatrix}^T \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\gamma & -s_\gamma \\ 0 & s_\gamma & c_\gamma \end{bmatrix}^T$$

$$\begin{bmatrix} c_\alpha & s_\alpha & 0 \\ -s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\beta & 0 & -s_\beta \\ 0 & 1 & 0 \\ s_\beta & 0 & c_\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\gamma & s_\gamma \\ 0 & -s_\gamma & c_\gamma \end{bmatrix}$$

$$\begin{bmatrix} c_\alpha & s_\alpha & 0 \\ -s_\alpha & c_\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\beta & s_\beta s_\gamma & -s_\beta c_\gamma \\ 0 & c_\gamma & s_\gamma \\ s_\beta & -c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix}$$

$$\begin{bmatrix} c_\alpha c_\beta & c_\alpha s_\beta s_\gamma + s_\alpha c_\gamma & -c_\alpha s_\beta c_\gamma + s_\alpha s_\gamma \\ -s_\alpha c_\beta & -s_\alpha s_\beta s_\gamma + c_\alpha c_\gamma & s_\alpha s_\beta c_\gamma + c_\alpha s_\gamma \\ s_\beta & -c_\beta s_\gamma & c_\beta c_\gamma \end{bmatrix}$$

Ésta última matriz de rotación es igual a la matriz de rotación de la trama S_6 , cuyos elementos los denominamos M_{ij} para coincidir con la nomenclatura utilizada de la implementación

$$\begin{bmatrix} M11 & M12 & M13 \\ M21 & M22 & M23 \\ M31 & M32 & M33 \end{bmatrix}$$

Despejando las variables α, β, γ :

$$\begin{aligned} -s_\alpha c_\beta &= M21 \\ c_\alpha c_\beta &= M11 \\ \Rightarrow \alpha &= \text{Atan2}(-M21, M11) \end{aligned}$$

$$\begin{aligned} s_\beta &= M31 \\ \Rightarrow \beta &= \text{sen}^{-1}(M31) \end{aligned}$$

$$\begin{aligned} -c_\beta s_\gamma &= M32 \\ c_\beta c_\gamma &= M33 \\ \Rightarrow \gamma &= \text{Atan2}(-M32, M33) \end{aligned}$$

Finalmente quedaría:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \text{Atan2}(-M21, M11) \\ \text{arcsen}(M31) \\ \text{Atan2}(-M32, M33) \end{bmatrix}$$

Sin embargo, la implementación del método anterior no fue satisfactoria.

Es importante notar que todos los resultados teóricos presentados tienen que transformarse al espacio de coordenadas particular de la implementación. En este caso, para DirectX y XNA, la dirección x en la implementación que se denominará X_XNA , equivale a la dirección z teórica, denominada $Z0_DH1$; la

dirección y (Y_{XNA}) equivale a la dirección z teórica (Z_{0_DH1}), y la dirección z (Z_{XNA}) equivale a la dirección x teórica (X_{0_DH1})

Esto se consigue aplicando una transformación extra, para obtener la alineación correcta entre ambos sistemas de coordenadas: una posibilidad es rotar primero en $Y_{0_DH1} - \pi/2$ radianes, para alinear Z_{0_DH1} con Y_{XNA} , y luego rotar de nuevo en el eje $Z_{0_DH1} - \pi/2$ radianes para alinear X_{0_DH1}' con Z_{XNA}

También, en la biblioteca XNA, los vectores se representan en forma renglón, y la matriz de la transformación tiene que transponerse y postmultiplicarse.

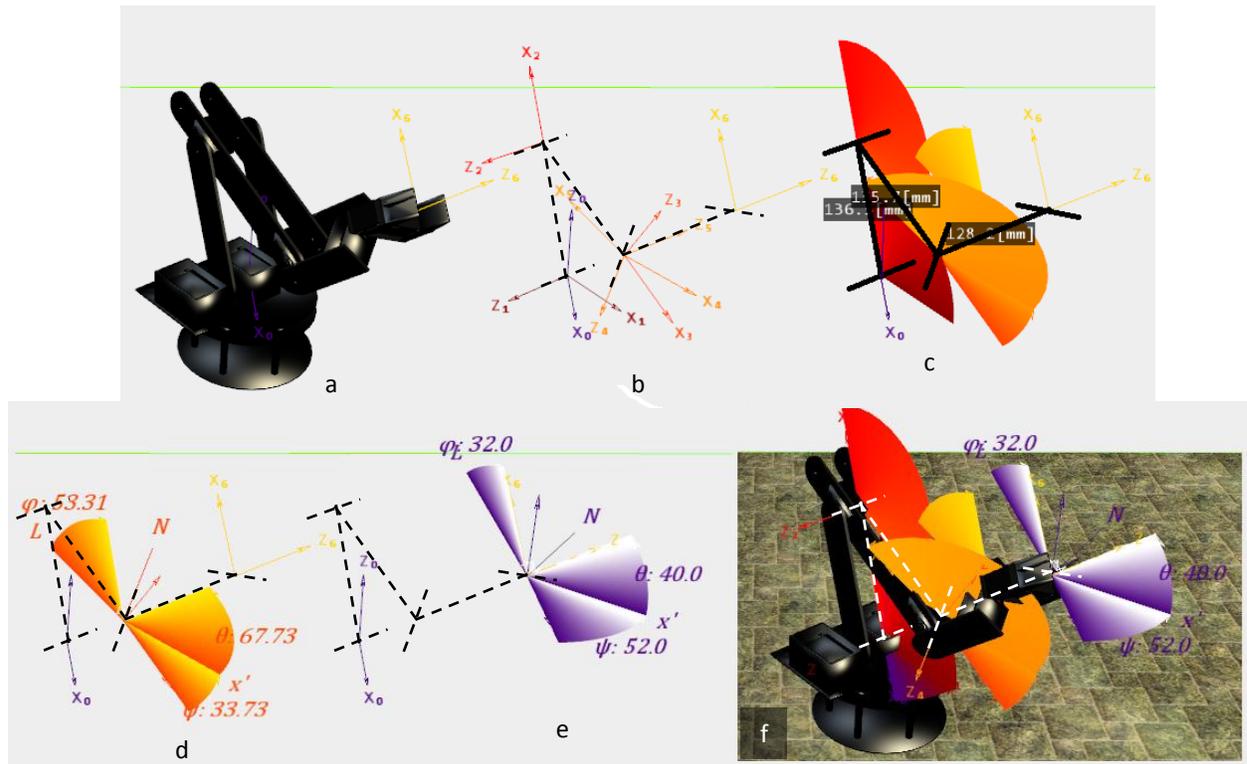


Figura 4.4. a) visualización tridimensional del robot. b) visualización de los ejes DH. c) visualización por sectores circulares para los parámetros angulares. d) valores para la orientación náutica de S_6 desde S_3 . e) valores para la orientación náutica de S_6 desde S_0 . f) Superposición de las capas de información anteriores.

5 Conclusiones

Estado final del proyecto

Se completó un sistema capaz de teleoperar y supervisar un manipulador prototipo en configuración serial de cinco grados de libertad. Este trabajo implica la teleoperación porque el control está dividido entre la estación de control y el prototipo físico. Más estrictamente, se trata de un sistema de control supervisado con una interfaz multimodal, para coincidir con la definición de Sheridan, aunque carece de telepresencia completa, por la ausencia de captura de video y audio provenientes del sitio operacional.

Para conseguir la teleoperación se implementó un protocolo de comunicación serial, se programó un microcontrolador esclavo y se desarrolló una interfaz multimodal.

Se desarrolló también una solución de cinemática inversa para seis grados de libertad, con el propósito de estudiar el desempeño cinemático de un diseño de estas características, ante distintos escenarios. A pesar de que el prototipo funcional fuese de cinco grados de libertad.

La interfaz cuenta con entorno virtual y distintas visualizaciones de las variables internas, las situacionales, y las de control y de comunicación para interactuar eficientemente con un operador. La plataforma virtual facilitó y añadió distintos modos de operación: directo, semiautomático inverso, restringido, asistido, seguimiento de trayectorias, evasión de obstáculos, etc. Posee también de infraestructura para la captura de datos provenientes de acelerómetros u otros dispositivos de instrumentación.

Durante la experimentación operativa, se encontró que la comunicación serial sufrió de intermitencias. El controlador puede optimizarse para volverse más robusto ante la presencia de errores en comunicación, incluso podría separarse de la interfaz y moverse al lado esclavo.

Cada servomotor posee su propio ciclo de control, aunque los parámetros no son observados por la interfaz.

Durante la implementación de las distintas clases, funciones y algoritmos de la plataforma virtual, (por ejemplo la deducción de orientación de Euler), queda claro que es necesario conocer en todo momento el estado del programa o sistema. Existen problemas que sólo se vuelven aparentes contando con alguna forma de visualización o realimentación visual, características que pertenecen a control supervisado. Es en los detalles donde se encuentra el mayor obstáculo en llevar la teoría a la práctica.

La forma de comprobar un algoritmo de cinemática inversa que previamente se desarrolló analíticamente y que después se implementó en el programa de forma numérica por medio de instrucciones, es por seguimiento de trayectorias o por manipulación directa. Cualquier discrepancia en el algoritmo se reflejó inmediatamente al observar que el extremo del modelo virtual no coincidía con el punto de operación.

Se probaron cuatro algoritmos distintos de cinemática inversa con resultados mixtos. Algunos resultados fueron correctos para algunas variables, y en otras ocasiones los resultados fueron inestables. Funcionaron algunas veces y otras no.

En su estado actual, experimentos con simulaciones en la plataforma virtual mostraron que tiene potencial para analizar configuraciones seriales redundantes, separando la cinemática inversa entre cadenas intermedias, que podría servir para analizar y operar robots paralelos, o robots móviles bajo el mismo principio. También se empezó con el estudio de algoritmos de manipulación como acoplamiento virtual y acercamiento.

Al sistema robótico pueden ocurrirle muchas cosas al momento de estarlo operando. Puede desconectarse un cable de alimentación, un cable de señal, puede recibir datos erróneos de comunicación serial (provocando sacudidas violentas), etc. De entre todas estas causas, ¿cómo se verifica que el algoritmo de cinemática inversa en uso produce resultados correctos? ¿Cómo se determina si el error en funcionamiento es por cinemática inversa u otros factores?

Para diagnosticar el estado operacional es necesario observar, monitorear o supervisar, las variables internas del robot. Esta es otra función fundamental de la plataforma virtual.

El que la plataforma sea virtual facilita la retroalimentación por visualización tridimensional, lo que brinda intuitividad a la hora de operar y hacer correcciones. El nombre de plataforma se empleó, porque la intención es de que sea una base sobre la cual se construyan algoritmos más sofisticados de control, supervisión, comunicación, visualización para alcanzar plenamente la teleoperación.

Se buscó que la plataforma fuese un lugar donde se realicen, capturen, analicen y visualicen y se pongan a prueba algoritmos o se diseñen estrategias de control o se programen secuencias de movimientos y trayectorias.

Al ser un ejercicio altamente interdisciplinario tal vez lo que convenga es desarrollar un proyecto similar en equipos multidisciplinarios, que incluyan ingenieros en telecomunicaciones, en computación, eléctricos, mecánicos para que aporten sus conocimientos y experiencia en las distintas tareas de diseño, relativos a cada componente del modelo de teleoperación.

Los sistemas teleoperados se caracterizan por ser pequeños e independientes en comparación con sistemas industriales. Una interfaz encuentra fuertes aplicaciones en ambos terrenos. El **enlace de comunicación** es un aspecto fundamental en teleoperación. La teleoperación se construye sobre telecomunicación. Es por esta razón que la teleoperación está muy ligada a paradigmas y tecnologías de telecomunicación. La interfaz en este caso cumple la función de acortar la distancia al presentar ante el usuario el entorno percibido por el robot, de forma integrada. Es ésta integración la que permite tomar decisiones de alto nivel, como identificación de la ubicación en el espacio, navegación, etc. El entorno presentado funciona como un enlace, un portal, una ventana a través de la cual el usuario extiende su voluntad hasta el robot.

La interfaz provee también un puente entre el dispositivo mecatrónico bajo control, el usuario, otros dispositivos y otras tecnologías de información. La facilidad con que se administra éste sistema complejo y variado, depende enteramente del diseño y las capacidades de la interfaz: que tan fácil es expedir comandos operativos, introducir datos (por ejemplo coordenadas articulares), visualizar algoritmos (por ejemplo la búsqueda de trayectorias en espacio de configuración), que tan fácil es capturar, analizar, consultar, transferir datos experimentales y operacionales, etc.

El Control y monitoreo de sistemas complejos y de diversas tecnologías debe de ser simple, intuitivo y ágil. La plataforma constituye parte de un producto tecnológico

Un sistema antes de ser automático (completamente autónomo) pasa por un proceso gradual de teleoperación y automatización. La interfaz siempre está presente la función de la interfaz pasa de operador, a supervisor

La interfaz integra y coordina distintas tecnologías

Aún quedaron muchas tareas pendientes por resolver, a pesar de ser de gran interés práctico. Pero que por restricciones de tiempo y costo, se decidió dejar aparte. Algunas de las extensiones o trabajo a futuro del presente proyecto, se enumeran a continuación:

- Visualización del espacio de configuración, visualización del jacobiano.
- Implementación de interpolación cúbica de trayectorias diferenciales en \mathbb{R}^6 usando cuaterniones
- Implementación de algoritmos de navegación paramétrica en espacio de configuración con obstáculos, inteligencia artificial, exploración, algoritmo A*.
- Integración de video de alta definición y micrófonos,
- Integración de pares de acelerómetros, magnetómetros, giroscopios, por grado de libertad.
- Implementación de comunicación inalámbrica con gran ancho de banda.
- Implementación de visión computarizada para determinar la postura operacional del robot, utilizando marcadores infrarrojos o transmisores de radiofrecuencia o ambos.
- Rediseño mecánico del manipulador: utilizar una transmisión para el primer grado de libertad, un mecanismo paralelo para el segundo y tercer grado de libertad, utilizar engranes diferenciales para

los últimos 3 grados de libertad, utilizar rodamientos adecuados, utilizar motores con encoders de efecto Hall o motores paso a paso.

- Rediseño del mecanismo de sujeción terminal (pinza),
- Integración de sensores de fuerza en cada articulación y el elemento terminal;
- Diseño e integración de un mecanismo de visión binocular desacoplado o multicámara, con suspensión de cardan redundante de cuatro grados de libertad, actuado.
- Obtención e implementación del modelo dinámico, para simulación y control robusto de siete grados de libertad
- Implementación de una interfaz secundaria en un dispositivo móvil o smartphone.
- Integración con un módulo cliente servidor para comunicación por Internet, o añadir funcionalidad de este tipo.

6 Apéndices

6.1 Desarrollo de ecuación matricial del modelo cinemático

Se obtienen 12 ecuaciones no lineales y trascendentes, que sirven para determinar la trama del elemento terminal S_6 respecto de la trama base S_0 en función del vector de coordenadas articulares $\mathbf{q} \in \mathbb{R}^6$. Ver sección 2.5, para una explicación de la simbología y nomenclatura. Los elementos de la matriz homogénea de la trama herramienta, quedan denotados por:

$${}^0T_6 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

De la Figura 2.11 de la sección 2.5, quedaron definidos los siguientes vectores de posición:

${}^5\mathbf{p}_6$	$[0,0,d_6]^T$	Posición del marco S_6 respecto del origen de S_5
${}^4\mathbf{p}_5$	$[0,0,0]^T$	Posición de S_5 respecto de S_4
${}^3\mathbf{p}_4$	$[0,0,0]^T$	Posición de S_4 respecto de S_3
${}^2\mathbf{p}_3$	$[a_3c_3, a_3s_3, 0]^T$	Posición de S_3 descrita desde S_2
${}^1\mathbf{p}_2$	$[a_2c_2, a_2s_2, 0]^T$	Posición de S_2 relativa a S_1
${}^0\mathbf{p}_1$	$[0,0,0]^T$	Posición de S_1 respecto S_0

Y también la siguiente tabla de matrices de rotación:

Tabla 6.1 Matrices de Rotación						
${}^{i-1}R_i$	0R_1	1R_2	2R_3	3R_4	4R_5	5R_6
$\begin{bmatrix} c_i & -s_i c_{\alpha_i} & s_i s_{\alpha_i} \\ s_i & c_i c_{\alpha_i} & -c_i s_{\alpha_i} \\ 0 & s_{\alpha_i} & c_{\alpha_i} \end{bmatrix}$	$\begin{bmatrix} c_1 & 0 & s_1 \\ s_1 & 0 & -c_1 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} c_3 & 0 & -s_3 \\ s_3 & 0 & c_3 \\ 0 & -1 & 0 \end{bmatrix}$	$\begin{bmatrix} c_4 & 0 & s_4 \\ s_4 & 0 & -c_4 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} c_5 & 0 & s_5 \\ s_5 & 0 & -c_5 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} c_6 & -s_6 & 0 \\ s_6 & c_6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
(α, θ)	$(\pi/2, \theta_1)$	$(0, \theta_2)$	$(-\pi/2, \theta_3)$	$(\pi/2, \theta_4)$	$(\pi/2, \theta_5)$	$(0, \theta_6)$

Queremos encontrar la posición del punto de operación P , situado en el origen de la trama S_6 , descrito desde del sistema raíz S_0 . P es el resultado de rotar y trasladar sucesivamente el origen y vector resultante de acuerdo a los desplazamientos previamente definidos, quedando lo siguiente:

$${}^0\mathbf{p}_6 = {}^0R_1({}^1R_2({}^2R_3({}^3R_4({}^4R_5({}^5\mathbf{p}_6 + {}^4\mathbf{p}_5) + {}^3\mathbf{p}_4) + {}^2\mathbf{p}_3) + {}^1\mathbf{p}_2) + {}^0\mathbf{p}_1$$

Desarrollando

$${}^0\mathbf{p}_6 = {}^0R_1 {}^1R_2 {}^2R_3 {}^3R_4 {}^4R_5 {}^5\mathbf{p}_6 + {}^0R_1 {}^1R_2 {}^2R_3 {}^3R_4 {}^4\mathbf{p}_5 + {}^0R_1 {}^1R_2 {}^2R_3 {}^3\mathbf{p}_4 + {}^0R_1 {}^1R_2 {}^2\mathbf{p}_3 + {}^0R_1 {}^1\mathbf{p}_2 + {}^0\mathbf{p}_1$$

Agrupando las matrices de rotación

$${}^0\mathbf{p}_6 = {}^0\mathbf{R}_5 {}^5\mathbf{p}_6 + {}^0\mathbf{R}_4 {}^4\mathbf{p}_5 + {}^0\mathbf{R}_3 {}^3\mathbf{p}_4 + {}^0\mathbf{R}_2 {}^2\mathbf{p}_3 + {}^0\mathbf{R}_1 {}^1\mathbf{p}_2 + {}^0\mathbf{p}_1$$

El segundo, tercer, y último término son cero, así que,

$${}^0\mathbf{p}_6 = {}^0\mathbf{R}_5 {}^5\mathbf{p}_6 + {}^0\mathbf{R}_2 {}^2\mathbf{p}_3 + {}^0\mathbf{R}_1 {}^1\mathbf{p}_2$$

Se encuentra primero ${}^0\mathbf{R}_2$:

$$\begin{aligned} {}^0\mathbf{R}_2 &= {}^0\mathbf{R}_1 {}^1\mathbf{R}_2 \\ {}^0\mathbf{R}_2 &= \begin{bmatrix} c_1 & 0 & s_1 \\ s_1 & 0 & -c_1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} c_2 & -s_2 & 0 \\ s_2 & c_2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ {}^0\mathbf{R}_2 &= \begin{bmatrix} c_1c_2 & -c_1s_2 & s_1 \\ s_1c_2 & -s_1s_2 & -c_1 \\ s_2 & c_2 & 0 \end{bmatrix} \end{aligned}$$

Después ${}^2\mathbf{R}_4$:

$$\begin{aligned} {}^2\mathbf{R}_4 &= {}^2\mathbf{R}_3 {}^3\mathbf{R}_4 \\ {}^2\mathbf{R}_4 &= \begin{bmatrix} c_3 & 0 & -s_3 \\ s_3 & 0 & c_3 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} c_4 & 0 & s_4 \\ s_4 & 0 & -c_4 \\ 0 & 1 & 0 \end{bmatrix} \\ {}^2\mathbf{R}_4 &= \begin{bmatrix} c_3c_4 & -s_3 & c_3s_4 \\ s_3c_4 & c_3 & s_3s_4 \\ -s_4 & 0 & c_4 \end{bmatrix} \end{aligned}$$

Después ${}^0\mathbf{R}_4$:

$$\begin{aligned} {}^0\mathbf{R}_4 &= {}^0\mathbf{R}_2 {}^2\mathbf{R}_4 \\ {}^0\mathbf{R}_4 &= \begin{bmatrix} c_1c_2 & -c_1s_2 & s_1 \\ s_1c_2 & -s_1s_2 & -c_1 \\ s_2 & c_2 & 0 \end{bmatrix} \begin{bmatrix} c_3c_4 & -s_3 & c_3s_4 \\ s_3c_4 & c_3 & s_3s_4 \\ -s_4 & 0 & c_4 \end{bmatrix} \end{aligned}$$

Primer Columna de ${}^0\mathbf{R}_4$:

$$\begin{array}{lll} c_1c_2c_3c_4 & -c_1s_2s_3c_4 & -s_1s_4 \\ s_1c_2c_3c_4 & -s_1s_2s_3c_4 & -c_1s_4 \\ s_2c_3c_4 & +c_2s_3c_4 & +0 \end{array}$$

Segunda Columna de ${}^0\mathbf{R}_4$:

$$\begin{array}{ll} -c_1c_2s_3 & -c_1s_2c_3 \\ -s_1c_2s_3 & -s_1s_2c_3 \\ -s_2s_3 & +c_2c_3 \end{array}$$

Tercer Columna de ${}^0\mathbf{R}_4$:

$$\begin{array}{lll} c_1c_2c_3s_4 & -c_1s_2s_3s_4 & s_1c_4 \\ s_1c_2c_3s_4 & -s_1s_2s_3s_4 & -c_1c_4 \end{array}$$

$$s_2c_3s_4 \quad +c_2s_3s_4$$

Luego 0R_5 :

$${}^0R_5 = {}^0R_4 {}^4R_5$$

$${}^0R_5 = {}^0R_4 \begin{bmatrix} c_5 & 0 & s_5 \\ s_5 & 0 & -c_5 \\ 0 & 1 & 0 \end{bmatrix}$$

Cuya primer columna es la combinación lineal de las columnas de 0R_4 , de acuerdo a los elementos de la primer columna de 4R_5 :

$$\begin{vmatrix} c_1c_2c_3c_4c_5 & -c_1s_2s_3c_4c_5 & -s_1s_4c_5 & -c_1c_2s_3s_5 & -c_1s_2c_3s_5 \\ s_1c_2c_3c_4c_5 & -s_1s_2s_3c_4c_5 & +c_1s_4c_5 & -s_1c_2s_3s_5 & -s_1s_2c_3s_5 \\ s_2c_3c_4c_5 & +c_2s_3c_4c_5 & & -s_2s_3s_5 & +c_2c_3s_5 \end{vmatrix}$$

La segunda columna de 0R_5 es idéntica a la tercera de 0R_4 :

$$\begin{matrix} c_1c_2c_3s_4 & -c_1s_2s_3s_4 & s_1c_4 \\ s_1c_2c_3s_4 & -s_1s_2s_3s_4 & -c_1c_4 \\ s_2c_3s_4 & +c_2s_3s_4 & \end{matrix}$$

La tercer columna de 0R_5 es la combinación lineal de las columnas de 0R_4 , de acuerdo a los elementos de la tercer columna de 4R_5 :

$$\begin{vmatrix} c_1c_2c_3c_4s_5 - c_1s_2s_3c_4s_5 - s_1s_4s_5 & -c_1c_2s_3c_5 & -c_1s_2c_3c_5 \\ s_1c_2c_3c_4s_5 - s_1s_2s_3c_4s_5 + c_1s_4s_5 & -s_1c_2s_3c_5 & -s_1s_2c_3c_5 \\ s_2c_3c_4s_5 + c_2s_3c_4s_5 & 0 & -s_2s_3c_5 + c_2c_3c_5 \end{vmatrix}$$

Obtenemos finalmente 0R_6

$${}^0R_6 = {}^0R_5 {}^5R_6$$

$${}^0R_6 = {}^0R_5 \begin{bmatrix} c_6 & -s_6 & 0 \\ s_6 & c_6 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$${}^0R_6 = \begin{bmatrix} n_x & s_x & a_x \\ n_y & s_y & a_y \\ n_z & s_z & a_z \end{bmatrix}$$

Quedando:

$$n_x = \begin{matrix} c_1c_2c_3c_4c_5c_6 - c_1s_2s_3c_4c_5c_6 - s_1s_4c_5c_6 - c_1c_2s_3s_5c_6 - c_1s_2c_3s_5c_6 \\ c_1c_2c_3s_4s_6 - c_1s_2s_3s_4s_6 + s_1c_4s_6 \end{matrix}$$

$$n_y = \begin{matrix} s_1c_2c_3c_4c_5c_6 - s_1s_2s_3c_4c_5c_6 + c_1s_4c_5c_6 - s_1c_2s_3s_5c_6 - s_1s_2c_3s_5c_6 \\ s_1c_2c_3s_4s_6 - s_1s_2s_3s_4s_6 - c_1c_4s_6 \end{matrix}$$

$$n_z = \begin{matrix} s_2c_3c_4c_5c_6 + c_2s_3c_4c_5c_6 - s_2s_3s_5c_6 + c_2c_3s_5c_6 \\ s_2c_3s_4s_6 + c_2s_3s_4s_6 \end{matrix}$$

$$s_x = \begin{matrix} -c_1c_2c_3c_4c_5s_6 + c_1s_2s_3c_4c_5s_6 + s_1s_4c_5s_6 + c_1c_2s_3s_5s_6 + c_1s_2c_3s_5s_6 \\ +c_1c_2c_3s_4c_6 - c_1s_2s_3s_4c_6 + s_1c_4c_6 \end{matrix}$$

$$\begin{aligned}
s_y &= -s_1c_2c_3c_4c_5s_6 + s_1s_2s_3c_4c_5s_6 - c_1s_4c_5s_6 + s_1c_2s_3s_5s_6 + s_1s_2c_3s_5s_6 \\
&\quad + s_1c_2c_3s_4c_6 - s_1s_2s_3s_4c_6 - c_1c_4c_6 \\
s_z &= -s_2c_3c_4c_5s_6 + -c_2s_3c_4c_5s_6 + s_2s_3s_5s_6 - c_2c_3s_5s_6 \\
&\quad + s_2c_3s_4c_6 + c_2s_3s_4c_6 \\
a_x &= c_1c_2c_3c_4s_5 - c_1s_2s_3c_4s_5 - s_1s_4s_5 + c_1c_2s_3c_5 + c_1s_2c_3c_5 \\
a_y &= s_1c_2c_3c_4s_5 - s_1s_2s_3c_4s_5 + c_1s_4s_5 + s_1c_2s_3c_5 + s_1s_2c_3c_5 \\
a_z &= s_2c_3c_4s_5 + c_2s_3c_4s_5 + s_2s_3c_5 - c_2c_3c_5
\end{aligned}$$

Una vez obtenidas las matrices de rotación, encontramos ${}^0\mathbf{p}_6$:

$$\begin{aligned}
{}^0\mathbf{p}_6 &= {}^0\mathbf{R}_5 {}^5\mathbf{p}_6 + {}^0\mathbf{R}_2 {}^2\mathbf{p}_3 + {}^0\mathbf{R}_1 {}^1\mathbf{p}_2 \\
{}^0\mathbf{p}_6 &= \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}
\end{aligned}$$

Quedando:

$$\begin{aligned}
p_x &= d_6(c_1c_2c_3c_4s_5 - c_1s_2s_3c_4s_5 - s_1s_4s_5 + c_1c_2s_3c_5 + c_1s_2c_3c_5) \\
&\quad + a_3(c_1c_2c_3 - c_1s_2s_3) + a_2c_1c_2 \\
p_y &= d_6(s_1c_2c_3c_4s_5 - s_1s_2s_3c_4s_5 + c_1s_4s_5 + s_1c_2s_3c_5 + s_1s_2c_3c_5) \\
&\quad + a_3(s_1c_2c_3 - s_1s_2s_3) + a_2s_1c_2 \\
p_z &= d_6(s_2c_3c_4s_5 + c_2s_3c_4s_5 + s_2s_3c_5 - c_2c_3c_5) \\
&\quad + a_3(s_2c_3 + c_2s_3) + a_2s_2
\end{aligned}$$

Otras nomenclaturas utilizadas:

$$\begin{aligned}
({}^0\mathbf{T}_6)^T &= \begin{bmatrix} n_x & n_y & n_z & 0 \\ s_x & s_y & s_z & 0 \\ a_x & a_y & a_z & 0 \\ p_x & p_y & p_z & 1 \end{bmatrix} \\
({}^0\mathbf{T}_6)^T &= \begin{bmatrix} M11 & M12 & M13 & 0 \\ M21 & M22 & M23 & 0 \\ M31 & M32 & M33 & 0 \\ M41 & M42 & M43 & 1 \end{bmatrix}
\end{aligned}$$

Resultados de cinemática directa obtenidos en Maple 17

$$\begin{aligned}
 n_x &= c_1 c_2 c_3 c_4 c_5 c_6 - c_1 c_4 c_5 c_6 s_2 s_3 + c_1 c_2 c_3 s_4 s_6 - c_1 c_2 c_6 s_3 s_5 - c_1 c_3 c_6 s_2 s_5 \\
 &\quad - c_1 s_2 s_3 s_4 s_6 - c_5 c_6 s_1 s_4 + c_4 s_1 s_6 \\
 n_y &= c_2 c_3 c_4 c_5 c_6 s_1 - c_4 c_5 c_6 s_1 s_2 s_3 + c_2 c_3 s_1 s_4 s_6 - c_2 c_6 s_1 s_3 s_5 - c_3 c_6 s_1 s_2 s_5 \\
 &\quad - s_1 s_2 s_3 s_4 s_6 + c_1 c_5 c_6 s_4 - c_1 c_4 s_6 \\
 n_z &= c_2 c_4 c_5 c_6 s_3 + c_3 c_4 c_5 c_6 s_2 + c_2 c_3 c_6 s_5 + c_2 s_3 s_4 s_6 + c_3 s_2 s_4 s_6 - c_6 s_2 s_3 s_5 \\
 s_x &= -c_1 c_2 c_3 c_4 c_5 s_6 + c_1 c_4 c_5 s_2 s_3 s_6 + c_1 c_2 c_3 c_6 s_4 + c_1 c_2 s_3 s_5 s_6 \\
 &\quad + c_1 c_3 s_2 s_5 s_6 - c_1 c_6 s_2 s_3 s_4 + c_5 s_1 s_4 s_6 + c_4 c_6 s_1 \\
 s_y &= -c_2 c_3 c_4 c_5 s_1 s_6 + c_4 c_5 s_1 s_2 s_3 s_6 + c_2 c_3 c_6 s_1 s_4 + c_2 s_1 s_3 s_5 s_6 \\
 &\quad + c_3 s_1 s_2 s_5 s_6 - c_6 s_1 s_2 s_3 s_4 - c_1 c_5 s_4 s_6 - c_1 c_4 c_6 \\
 s_z &= -c_2 c_4 c_5 s_3 s_6 - c_3 c_4 c_5 s_2 s_6 - c_2 c_3 s_5 s_6 + c_2 c_6 s_3 s_4 + c_3 c_6 s_2 s_4 \\
 &\quad + s_2 s_3 s_5 s_6 \\
 a_x &= c_1 c_2 c_3 c_4 s_5 - c_1 c_4 s_2 s_3 s_5 + c_1 c_2 s_5 s_3 + c_1 c_3 c_5 s_2 - s_1 s_4 s_5 \\
 a_y &= c_2 c_3 c_4 s_1 s_5 - c_4 s_1 s_2 s_3 s_5 + c_2 c_5 s_1 s_3 + c_3 c_5 s_1 s_2 + c_1 s_4 s_5 \\
 a_z &= c_2 c_4 s_3 s_5 + c_3 c_4 s_2 s_5 + c_2 c_3 c_5 - c_5 s_2 s_3
 \end{aligned}$$

Para comprobar que efectivamente n , s y a son ortogonales entre sí, y de longitud unitaria, su producto punto debe de ser cero:

$$n \cdot a = 0$$

$$n \cdot s = 0$$

$$a \cdot s = 0$$

Pero dada la complejidad algebraica de las operaciones anteriores, se presenta solamente como ejercicio para el interesado

6.2 Programa final de comunicación y control con el prototipo

```
//=====
// BotProtocol v0.2 |
// by Daniel Ortiz Pámanes |
// white_pawn1@hotmail.com |
// Late Winter 2013 |
//=====

#include <Servo.h>

const boolean DEBUG_SERIAL = false; // WriteLedBinary
const boolean OFFLINE_DEMO = true; // debug Sweep
const byte BINARY_MODE = 0x00; //B00000000
byte MODE;

const byte HI = 0xF9; // f_00E
const byte CONTROL_CHARACTER = 0xFA; // f_10A
const byte PACKET_CONFIRMED = 0xFB; // f_200
const byte RESEND_PACKET = 0xFC; // f_204
const byte DISCARD = 0xFD; // f_204
const byte SAY_WHO_YOU_ARE = 0xFE; // f_00F

// Holds decoded data from serial port
float ReceivedConfig[6];
float CurrentConfig[6];
float HomeConfig[] = { 80, 157, 178, 172, 68, 0 };
float VerticalConfig[] = { 80, 80, 38, 105, 68, 0 };
float VerticalConfig_01[] = { 178, 80, 38, 105, 178, 90 };
float TestConfig_00[] = { 60, 84, 38, 0, 178, 70 };
float TestConfig_01[] = { 60, 84, 0, 180, 0, 20 };
float TestConfig_02[] = { 80, 18, 38, 105, 68, 0 }; // Horizontal
float TestConfig_03[] = { 0, 157, 178, 37, 0, 0 }; // Close to Home
float TestConfig_04[] = { 178, 157, 178, 37, 178, 0 }; // Close to Home2
float TestConfig_05[] = { 80, 80, 38, 105, 68, 90 };

//
// _____ |
// M A I N |
// _____ |

boolean ReadAllowed = false; // If data arrives correctly, becomes true
void setup()
{
  MODE = BINARY_MODE;
  SetupLedPins(); // Displays binary data of a byte of received data packet
  SetupServos();
  if(OFFLINE_DEMO)
  {
    BlinkCode_00();
  }
  else
  {
    BlinkCode_01();
  }
}
```

```

    // SetupLed(); // Onboard led pin 13
}
CopyArray(VerticalConfig, CurrentConfig); // Critical
WriteServos(CurrentConfig);
Serial.begin(9600); // {115200, 9600}
Serial.setTimeout(5);
}
void loop()
{
  if(OFFLINE_DEMO)
  {
    //Demo();
    Demo_II(); // Testing Gripper
  }
  else
  {
    ParseControl(false); // Decodes Control character
    if(ReadAllowed)
    {
      switch(MODE)
      {
        case BINARY_MODE:
          ParseBinary();
        }
      //TestParsing(ReceivedConfig[0]);
      ReadAllowed = false;
    }
    Sweep(ReceivedConfig, false);
  }
}
void BlinkCode_00()
{
  // for OFFLINE_DEMO
  Blink(100);
  Blink(100);
  Blink(100);
  Blink(300);
  Blink(300);
  Blink(300);
  Blink(100);
  Blink(100);
  Blink(100);
}
void BlinkCode_01()
{
  // for ONLINE / LIVE
  Blink(200);
  Blink(200);
  Blink(200);
  Blink(200);
}

```

```

// Filename: "Serial.INO"
//
// INDEX
//
// 000. SetupLed()
// 001. Blink()
// 002. SetupLedPins()
// 003. IntToBinary()
// 004. WriteLedBinary()
// 005. TestParsing()
// 006. ParseControl()
// 007. DecodeByte()
// 008. ParseBinary()
// 009. RequestMoreData()
// 00A. ResendRequest()
//

//
// SetupLed
//
void SetupLed()
{
    pinMode(13, OUTPUT);
}

//
// Blink
//
void Blink(int value)
{
    digitalWrite(13, HIGH);
    delay(value/2);
    digitalWrite(13, LOW);
    delay(value/2);
}

//
// SetupLedPins
//
short LedPins[] = {13,12,8,7,4,2}; // LSB <--> MSB
// Pins 3, 5, 6, 9, 10, 11 are used for PWM
// only six bits available for decoding
void SetupLedPins()
{
    for(int i = 0; i < 8; i++)
        pinMode(LedPins[i], OUTPUT);
}

//
// IntToBinary
//

```

```

boolean LedArrayOnOff[8] = {false, true, false, true, false, true, false, true};
void IntToBinary(int value)
{
  LedArrayOnOff[0] = (value & B00000001) > 0; // 0, 1
  LedArrayOnOff[1] = (value & B00000010) > 0; // 0, 2
  LedArrayOnOff[2] = (value & B00000100) > 0; // 0, 4
  LedArrayOnOff[3] = (value & B00001000) > 0; // 0, 8
  LedArrayOnOff[4] = (value & B00010000) > 0; // 0, 16
  LedArrayOnOff[5] = (value & B00100000) > 0; // 0, 32
  LedArrayOnOff[6] = (value & B01000000) > 0; // 0, 64
  LedArrayOnOff[7] = (value & B10000000) > 0; // 0, 128
}

// _____
// |                                     |
// |   WriteLedBinary                   |
// |_____                             |
void WriteLedBinary()
{
  for (int i =0; i<6; i++)
  {
    if(LedArrayOnOff[i])
      digitalWrite(LedPins[i], HIGH);
    else
      digitalWrite(LedPins[i], LOW);
  }
}

// _____
// |                                     |
// |   TestParsing                       |
// |_____                             |
// Alternative names: PrintBinary
void TestParsing(int value)
{
  // "SetupLedPins()" must be called first
  int testValue;
  // Examples:
  // testValue = B11001100;
  // testValue = parsedAngle[0][index];
  // testValue = Serial.available() & B00001000
  // if(testPassed)
  // donextTest()
  IntToBinary(value);
  WriteLedBinary();
}

// _____
// |                                     |
// |   ParseControl                      |
// |_____                             |
// Alternative names DecodeControl
void ParseControl(boolean debug)
{
  if(Serial.available() > 6)

```

```

{
  byte inByte = Serial.read();
  if(debug)
    TestParsing(inByte); // Decodes inbyte to Binary and Writes to Leds
  if(DecodeByte(inByte, CONTROL_CHARACTER))
  {
    RequestMoreData();
    ReadAllowed = true;
    return;
  }
  RequestResend(false);
  Serial.flush();
}
}

// _____ |
// ParseBinary |
// _____ |
// Alternative names: DecodeBinary, ByteToInt, ParseByte, ParseInt
void ParseBinary()
{
  for (int i = 0; i<6; i++)
  {
    ReceivedConfig[i] = Serial.read();
  }
}

// _____ |
// RequestMoreData |
// _____ |
void RequestMoreData()
{
  Serial.write(PACKET_CONFIRMED);
}

// _____ |
// ResendRequest |
// _____ |
int resendsLeft = 1;
void RequestResend(boolean DiscardON)
{
  if(resendsLeft > 0)
  {
    Serial.write(RESEND_PACKET);
    resendsLeft--;
  }
  else if (DiscardON)
  {
    Serial.write(DISCARD);
    resendsLeft = 1;
  }
}
}

```

```

// " Servo.INO "
//
// INDEX
// 100. SetupServos()
// 101. WriteServos()
// 102. Demo()
// 103. Sweep()
//

//
//
// SetupServos
//
Servo S[6]; // {Base, Hombro, Codo, Pitch, Roll, Pinza}
int ServoPin[] = {
  3,5,6,9,10,11};
void SetupServos()
{
  S[0].attach(ServoPin[0]);
  S[1].attach(ServoPin[1]);
  S[2].attach(ServoPin[2]);
  S[3].attach(ServoPin[3]);
  S[4].attach(ServoPin[4]);
  S[5].attach(ServoPin[5]);
}

//
//
// WriteServos
//
void WriteServos(float Config[6])
{
  S[0].write(Config[0]);
  S[1].write(Config[1]);
  S[2].write(Config[2]);
  S[3].write(Config[3]);
  S[4].write(Config[4]);
  S[5].write(Config[5]);
}

```

```

//_____
//      |
//      Demo      |
//_____
void Demo()
{
  Sweep(   TestConfig_01, false);
  Sweep(   VerticalConfig, false);
  Sweep(   TestConfig_02, false);
  Sweep(   TestConfig_04, false);
  Sweep(   HomeConfig, false);
  Sweep(   TestConfig_03, false);
  Sweep(VerticalConfig_01, false);
  Sweep(   TestConfig_00, false);
}
void Demo_II()
{
  Sweep(VerticalConfig, false);
  Sweep(VerticalConfig_01, false);
}

//_____
//      |
//      Sweep      |
//_____
void Sweep(float TargetConfig[6], boolean debug)
{
  int timeDelay = 10;
  float DeltaConfig[6]; // DeltaConfig = TargetConfig - CurrentConfig
  CopyArray(Difference(TargetConfig, CurrentConfig), DeltaConfig); //
DeltaConfig = TargetConfig - CurrentConfig
  float d = Distance(DeltaConfig);
  float steps = 0.9f * d;
  if(debug)
  {
    Serial.print("d: ");
    Serial.println(steps);
  }
  if(d > 1)
  {
    CopyArray(ScalarMultiply(1/steps, DeltaConfig), DeltaConfig); // DeltaConfig
/= steps
    for(int t = 0; t < steps; t++)
    {
      CopyArray(Sum(CurrentConfig, DeltaConfig), CurrentConfig); //
CurrentConfig += DeltaConfig;
      WriteServos(CurrentConfig);
      //SerialPrint(CurrentConfig);
      //Blink(timeDelay);
      delay(timeDelay);
      TestParsing(CurrentConfig[0]);
    }
  }
}
else

```

```

    {
        CopyArray(TargetConfig, CurrentConfig);
        WriteServos(CurrentConfig);
        TestParsing(CurrentConfig[0]);
    }
}

// Filename: "Vector6.INO"
// _____
//      INDEX
// _____
// 200. CopyArray()
// 201. Sum()
// 202. Difference()
// 203. Distance()
// 204. ScalarMultiply()
// 205. SerialPrint()
// _____

// _____
//      CopyArray
// _____
void CopyArray(float from[6], float to[6])
{
    for(int i = 0; i < 6; i++)
        to[i] = from[i];
}

// _____
//      Sum
// _____
float* Sum(float Q1[6], float Q2[6])
{
    float result[6];
    for (int i = 0; i < 6; i++)
        result[i] = Q1[i] + Q2[i];
    return result;
}

// _____
//      Difference
// _____
float* Difference(float Q1[6], float Q2[6])
{
    float result[6];
    for (int i = 0; i < 6; i++)
        result[i] = Q1[i] - Q2[i];
    return result;
}

```

```

// _____
//          |
//   Distance |
// _____
float Distance(float Q[6])
{
    float d;
    for (int i = 0; i < 6; i++)
        d += Q[i]*Q[i];
    return sqrt(d);
}

// _____
//          |
//   ScalarMultiply |
// _____
float* ScalarMultiply(float scalar, float Q[6])
{
    float result[6];
    for(int i = 0; i < 6; i++)
        result[i] = scalar * Q[i];
    return result;
}

// _____
//          |
//   SerialPrint |
// _____
void SerialPrint(float Q[6])
{
    for(int i = 0; i < 6; i++)
    {
        Serial.print(Q[i]);
        Serial.print('\t');
    }
    Serial.println();
}

```

- [1] R.C. Arkin (Ed.) "Principles of Robot Motion: Theory, Algorithms, and Implementation" MIT Press, Englad, 2005.
- [2] G.H. Ballantyne, "Robotic Surgery, telerobotic surgery, telepresence and telemonitoring", Surgical Endoscopy, vol. 16, pp. 1389-1402, N.Y. 2002.
- [3] J.E. Bares, D.S. Wettergreen, "Dante II: Technical Description, Results and Lessons Learned", International Journal of Robotics Research, Vol. 18, No. 7, 1999.
- [4] A. Barrientos, et al. "Fundamentos de Robotica", 1ª Ed. McGraw-Hill, España, 1997.
- [5] P.G. Barros, R.W. Lindeman "A Survey of User Interfaces for Robot Teleoperation", Worcester Polytechnic Institute, 2008.
- [6] L. Basañez, R. Suárez, "Teleoperation", in "Springer Handbook of Automation", 2009.
- [7] C. Basdogan, "From 2D Images to 3D Tangible Models: Autostereoscopic and Haptic Visualization of Martian Rocks in Virtual Environments", Presence, Vol. 16, No. 1, 2007.
- [8] R.H. Bishop, "The Mechatronics Handbook", 2nd Edition, 2002.
- [9] W. Book, L. Love "Teleoperation telerobotics telepresence" in "Handbook of Industrial Robotics", 2nd Ed. Wiley, pp. 167-186. 2009.
- [10] J.J. Craig, "Robotica", 3a ed. Pearson, 2006.
- [11] J. Cui, S. Tosunoglu, R. Roberts, C. Moore, D.W. Repperger "A Review of Teleoperation system control", Proceedings of Florida Conference on Recent Advances in Robotics, FCRAR, 2003.
- [12] H. Flemmer, B. Eriksson and J. Wikander, "Control Design for Transparent Teleoperators with Model Parameter Variation", Proc. IEEE International Conference on Robotics & Automation, 2002
- [13] T. Fong, C. Thorpe, "Vehicle Teleoperation Interfaces", Autonomous Robots, Vol. 11, pp. 9-18, Kluwer Academic Publishers, Netherlands, 2001.
- [14] M. Fontolan "Prestige oil recovery from the sunken part of the Wreck", PAJ oil spill Symp. Petroleum Association of Japan, Tokyo, 2005.
- [Fu] K.S. Fu, R.C. Gonzalez, "Robotica: Control, detección, visión e inteligencia" McGraw-Hill 1988.
- [15] R. Garcia, J. Battle, X. Cufi, J. Amat, "Positioning and Underwater Vehicle through Image Mosaicking", Proc. International Conference on Robotics and Automation, 2001.
- [16] M.A. Goodrich, A.C. Shultz, "Human-Robot Interaction: A survey", in "Foundations and Trends in Human-Computer Interaction", vol. 1, No. 3, pp. 203-275, 2007.
- [17] R. Isermann, "Mechatronic Systems - A Short Introduction" in "Springer Handbook of Automation", pp. 317-331, 2009.
- [18] T.R. Kurfess (Ed.) "Robotics and Automation Handbook", CRC Press, 1st ed., 2005.
- [19] E. Lengyel "Mathematics for 3D game programming", 2nd Ed., 2004.
- [20] S. Lichiardopol "A Survey on Teleoperation" Technische Universiteit Eindhoven, Department Mechanical Engineering, 2007.
- [21] C.W. Nielsen, M.A. Goodrich, "Comparing the Usefulness of Video and Map Information in Navigation Tasks", 2006. (Conclusions)
- [22] S.Y. Nof, (Ed.) "Springer Handbook of Automation", Springer 2009.
- [23] Shimon T. Nof, "Handbook of Industrial robotics"
- [24] A.J. Park, R.N. Kazman, "Augmented reality for mining teleoperation" Proceeding of SPIE International Symposium of Intelligent Systems Adv. Manuf. – Telemanipulator and Telepresence Technologies. pp. 119-129, 1995. (Abstract)
- [25] F.E. Reyes Pinner, "Control Dinámico de un Manipulador Redundante de 7 GDL en Espacio Cartesiano usando Cuaterniones", Tesis Maestría Mecánica-Mecatrónica, UNAM, 2012.
- [26] G. Riemer, "XNA 3.0 Game Programming recipes, a problem solution approach", 2009.
- [27] ROTEX http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-3827/5969_read-8744/ (Consultado 2014.02.08)
- [28][Sheridan_92] T.B. Sheridan, "Telerobotics, Automation and Human Supervisory Control", MIT Press, Cambridge, 1992.
- [29][Sheridan_93] T.B. Sheridan, "Space Teleoperation Through Time Delay: Review and Prognosis" IEEE Trans. on Robotics and Automation, vol. 9, no. 5, 1993. (Abstract)

- [30][Sheridan_95] T.B. Sheridan, "Teleoperation, Telerobotics and Telepresence: A Progress Report", Control Engineering Practice, Vol.3, No. 2, pp. 204-214, 1995. (Abstract)
- [31] K. Shoemake, "Animating rotation with quaternion curves" SIGGRAPH, vol. 19, no. 3, 1985.
- [32] B. Siciliano, O. Khatib, (Eds.) "Springer Handbook of Robotics", Springer, 2008.
- [33] M.W. Spong, "Robot Dynamics and Control", 1989.
- [34] K. M. Stannery (Ed.), "Handbook of virtual environments", Mahwah, NJ: Erlbaum.
- [35] N. Tesla "Method of and Apparatus for controlling mechanism of moving vessels or vehicles", US patent 613,809, 1898.
- [36] A. Tobergte, R. Konietschke, G. Hirzinger "Planning and Control of a Teleoperation System for Resarch in Minimally Invasive Robotic Surgery", IEEE International Conference on Robotics and Automation, pp. 4225-4232, 2009.
- [37] Lung-Wen Tsai, "Robot Analysis: the mechanics of serial and parallel manipulators", Wiley & Sons, 1999.
- [38] D. Weijers, N. Geldner, R. Offringa. "Transatlantic robot-assisted telesurgery" Nature Vol. 416, 2001.
- [39] L.L. Whitcomb "Underwater Robotics: Out of the Research Laboratory and Into the Field", IEEE International Conference on Robotics and Automation, 2000.
- [40] J. Yuh "Design and Control of Autonomous Underwater robots: A Survey" Autonomous Robots 8, 7-24, 2000.