Resumen

La robótica es una técnica que estudia el diseño y la construcción de máquinas capaces de desempeñar tareas para el ser humano. Potencialmente, estas maquinas deberán trabajar en entornos y sobre objetos pensados para ser detectados y manipulados por los seres humanos. Por tanto, es lógico pensar que resulta de utilidad dotar a estas máquinas de ciertas aptitudes humanas como la visión, o de cualidades como la morfología de los manipuladores humanos.

El objetivo de este proyecto consiste en incorporar al sistema Mano-mecánica / Robot del Instituto de Organización y Control de Sistemas Industriales de la UPC (IOC), un sistema de visión por computador que permita la localización de objetos en el entorno de trabajo y su posterior manipulación. Este sistema integrado deberá servir de plataforma para probar y desarrollar teorías de localización, aprehensión y manipulación de objetos.

Con el propósito de integrar visión al sistema, se ha equipado el laboratorio de robótica del IOC con un ordenador, una cámara de video y una tarjeta de adquisición de imágenes. También se ha diseñado un software que integra el procesado de las imágenes, el cálculo de consignas y la comunicación con la Mano y el Robot. Asimismo, se ha adaptado el software de la Mano y el Robot para la recepción de consignas.

El sistema implantado permite realizar tareas de localización de objetos con eficiencia, permitiendo distinguir objetos por cualidades como la forma o el color, para posteriormente manipularlos. Así pues, se ha establecido una plataforma de trabajo en el IOC que permitirá experimentar, probar y desarrollar teorías de localización, aprehensión y manipulación de objetos.

Finalmente, cabe afirmar que los sistemas robotizados con visión por computador pueden ser utilizados como herramientas útiles en máquinas industriales o de servicios para realizar tareas complejas.

Sumario

RESUMEN 1

SUMARIO 3

1. GLOSARIO 5

2. INTRODUCCIÓN 9

2.1. Objetivo del proyecto 9

2.2. Alcance del proyecto 9

3. DESCRIPCIÓN DEL SISTEMA MR 11

3.1. Mano-mecánica 11

3.1.1. Mecánica de la mano 11

3.1.2. Hardware de Control 14

3.1.3. Software de la mano 16

3.2. Brazo Robot 18

3.2.1. Brazo (TX-90) 19

3.2.2. Armario de Control (CS8C) 22

3.2.3. Unidad de mando manual (MCP) 25

3.2.4. Entorno para el desarrollo de aplicaciones VAL3 (SRS) 28

3.2.5. Célula de trabajo y protocolos de actuación 28

4. DISEÑO E IMPLANTACIÓN DEL SISTEMA VMR 33

4.1. Requerimientos 33

4.2. Recursos 33

4.2.1. PC (COBE) 34

4.2.2. Tarjeta de adquisición de vídeo (Angelo RTV-24) 34

4.2.3. Cámara de vídeo (FCB-IX45CP) 35

4.2.4. Instalación hardware 36

4.3. Software del sistema VMR 38

4.3.1. Esquema funcional 38

4.3.2. Captura y tratamiento de imágenes 39

4.3.3. Determinación de consignas 41

4.3.4. Comunicación con el sistema MR 48

4.3.5. Interacción con el usuario 56

4.3.6. Procedimiento para la implantación de una tarea en VMR 63

5. APLICACIÓN DEL SISTEMA VMR DESARROLLADO ____ 67
5.1. Descripción de la tarea 67
5.2. Diseño e implementación 68
5.3. Experimentación 76
6. EVALUACIÓN DE IMPACTO AMBIENTAL ____ 83
7. PRESUPUESTO ____ 85
CONCLUSIONES ____ 89
AGRADECIMIENTOS ____ 91
BIBLIOGRAFÍA ____ 93
Referencias bibliográficas 93
Bibliografía complementaria 93

1. Glosario

C++: Lenguaje informático que abarca la programación estructurada, la programación genérica y la programación orientada a objetos.

C: Lenguaje informático creado en el 1969, utilizado en la implementación de numerosos sistemas operativos

COBE: PC utilizado para la integración de visión en el sistema Mano-mecánica / Robot.

CS8C: Armario de control del robot del IOC.

ESFERA: Aplicación experimental del sistema VMR en que se localiza una esfera de diámetro variable en el entorno de trabajo, se aprehende y se deposita en el lugar determinado por el usuario.

FTP: Siglas en inglés de File Transfer Protocol.

GCC Siglas en inglés de Colección de Compiladores GNU.

GDL: Grados de libertad.

GPL: Siglas en inglés de Licencia Pública General.

HTS: Siglas de la transformación geométrica HandToSphere.

HTW: Siglas de la transformación geométrica HandToWrist.

IOC: Instituto de Organización y Control de Sistemas Industriales de la UPC.

Dirección IP: Número que identifica en Internet a un ordenador que utiliza el protocolo IP (Internet Protocol).

LED: Diodo emisor de luz.

Lista FIFO: Lista en la que el primero en entrar es el primero en salir. (Siglas en Internet de First In First Out).

MA-I: Mano-mecánica antropomórfica de cuatro dedos utilizada en el IOC.

MAI:	Biblioteca de la cinemática directa de la mano desarrollada para el proyecto.
MAIQNX6:	Software desarrollado por el IOC para la mano MA-I.
MAIQNX6 Server:	Software desarrollado conjuntamente con el IOC para la recepción de consignas a través de Internet de la mano MA-I.
MCP:	Unidad de mando manual del brazo del IOC.
MR:	Sistema robotizado MA-I / Robot.
MT:	Biblioteca matemática desarrollada en el IOC que utiliza la librería de código abierto BOOST.
NTSC:	Siglas en inglés del sistema estadounidense de codificación de señales de televisión analógica en color (National Television System Comité).
PAL:	Siglas en inglés del sistema de codificación de señales de televisión analógica en color (Phase Alternating Line).
Sistema Háptico:	Interfaz que interacciona con el ser humano mediante el sentido del tacto.
STÄUBLI TX-90:	Brazo robot del IOC.
STH:	Siglas de la transformación geométrica SphereToHand.
TCP:	Siglas en inglés del Protocolo de Control de la Transmisión ampliamente utilizado en Internet.
TXRobot:	Biblioteca de la cinemática directa e inversa del robot que utiliza la librería MT, y que ha sido desarrollada por el IOC.
UDP:	Siglas en inglés del Protocolo de Datagramas de Usuario ampliamente utilizado en Internet.
VMR:	Sistema integrado de Visión, MA-I y Robot.
VMR VAL3:	Software desarrollado conjuntamente con el IOC para la recepción de consignas a través de Internet del robot TX-90.

Widget: En la terminología utilizada por Qt, se refiere a un contenedor de elementos gráficos.

WTSE: Siglas de la transformación geométrica WorldToSphereEntrega.

WTSR: Siglas de la transformación geométrica WorldToSphereRecogida.

WTP: Siglas de la transformación geométrica WorldToPhoto.

2. Introducción

La necesidad de integrar visión por computador en el sistema Mano / Robot (MR) del IOC nace de la falta de un entorno que unifique la captura y procesado de imágenes con el comando del sistema MR.

2.1. Objetivo del proyecto

El objetivo principal del proyecto es integrar visión por computador en el sistema MR para proporcionar una plataforma de pruebas experimentales de diferentes algoritmos y estrategias de localización, aprehensión y manipulación de objetos.

El control de los movimientos de la Mano y del Robot se realiza por separado, y por tanto es necesario dotar al sistema integrado de visión, Mano y Robot (VMR) de un entorno que permita comandar los dos sistemas.

El software resultante del proyecto debe ser multiplataforma debido al uso de diversos sistemas operativos en los equipos del IOC. Asimismo, deberá contar con herramientas integradas de procesado de imágenes, de cálculo (para la cinemática del Robot y de la Mano), métodos de comunicación entre equipos y una interfase gráfica con el usuario.

2.2. Alcance del proyecto

El VMR constará de una sola cámara para la adquisición de imágenes, situada en una posición fija respecto al Robot, pudiendo ser fácilmente ampliable hasta cuatro cámaras en total (permitiendo así la introducción en el sistema de técnicas de reconstrucción 3D).

La localización de objetos se realizará de forma estática, quedando fuera del ámbito de trabajo el análisis de movimiento y el seguimiento de objetos.

El control de posición, velocidad y fuerza tanto de la Mano como del Robot no se incluye en el VMR quedando circunscrito a sus respectivos controladores, dado que este sistema es de alto nivel.

3. Descripción del sistema MR

En este capítulo se realiza una descripción del sistema MR existente en el IOC en el que se realizará la integración de visión por computador.

3.1. Mano-mecánica

La Mano mecánica denominada MA-I (Mano Artificial Inteligente) diseñada y construida en el IOC es una mano antropomórfica, mecánicamente autocontenida y de diseño modular (ver Figura 3.1). A continuación se describe la mecánica de la mano, el hardware de control y el software del control e interacción de MA-I.

Figura 3.1 – Mano mecánica MA-I.

3.1.1. Mecánica de la mano

El peso de MA-I es aproximadamente de 4kg y sus dimensiones (con los dedos extendidos) son: 36cm de altura, 25cm de ancho y 30cm de profundidad. Los dedos están formados por piezas mecanizadas de aluminio, mientras que la estructura principal está construida en fundición y corresponde a la palma de la mano humana. Esta estructura aloja en su base un adaptador, permitiendo su ensamblaje al portaherramientas de un robot, y en el anverso se encuentran las conexiones con la etapa de control y de potencia.

La mano MA-I conserva las proporciones, la ubicación y la orientación de los dedos de una mano humana, la mayor diferencia radica en que tiene cuatro dedos en vez de cinco. La

reducción de un dedo se justifica al tener en cuenta que la inclusión del quinto dedo sólo incrementa un 5% las habilidades prensiles de una mano y sin embargo tiene un coste proporcionalmente mucho mayor [Suárez y Grosch, 2003, p.3].

Los dedos de MA-I se dividen en dos tipos según la longitud de las falanges que los componen. Los dedos anular, corazón e índice son exactamente iguales y pertenecen al Tipo I, mientras que el pulgar es de Tipo II. La longitud de las falanges se muestra en la Tabla 3.1.

Longitud falanges [mm]	Proximal	Medial	Distal
Tipo I	76.66	56.00	33.62
Tipo II	76.66	66.00	39.17

Tabla 3.1 – Longitud de las falanges para los Tipos I y II

En cuanto a la movilidad de los dedos cabe destacar que cada uno tiene cuatro grados de libertad (GDL). Al igual que en las manos humanas, un GDL permite realizar los movimientos de abducción y aducción, mientras que los tres GDL restantes permiten la extensión y la flexión de los dedos (Figura 3.2).

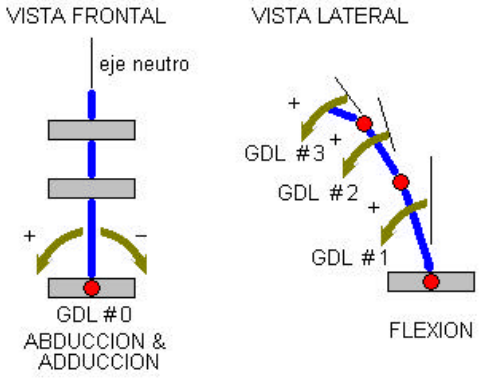


Figura 3.2 – Esquema de la movilidad de un dedo. Fuente: IOC

La mano se sirve de 16 articulaciones para moverse, 12 son de tipo bisagra y posibilitan los GDL #2 y #3 de cada dedo (Figura 3.2). Estas articulaciones se mueven mediante un motor-reductor DC conectado a un tornillo sinfín que a su vez engrana con una corona solidaria a la falange accionada (ver Figura 3.3).

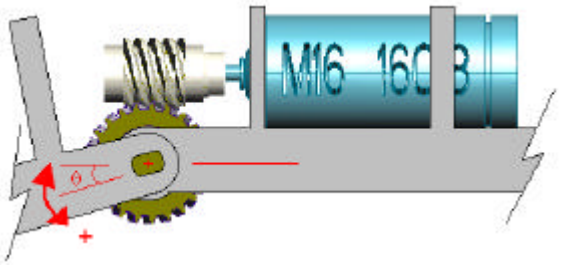


Figura 3.3 - Engranaje tipo bisagra. Fuente: IOC.

En cuanto a la mecánica de las transmisiones sinfín-corona, cabe destacar que reducen la velocidad de salida y aumentan el par, permitiendo realizar grandes esfuerzos. El único inconveniente de este tipo de articulaciones es que el sentido de giro es irreversible y la corona no puede actuar sobre el tornillo sinfín. Este giro sería útil si se pretende conocer el par que realiza la falange de salida sobre el motor.

Las cuatro articulaciones restantes son las que mueven la falange proximal de los cuatro dedos y permiten el movimiento de los GDL #0 y #1 descritos en la Figura 3.2. El GDL #0 realiza la abducción y la aducción del dedo sobre un eje situado en el plano medio longitudinal del dedo y que pasa por el centro de la base. El GDL #1 está acoplado al GDL #0 y se utiliza para la flexión y la extensión del dedo. El eje de este GDL mantiene un ángulo de 90º en todo momento con el eje del GDL #0 y pertenece al plano base del dedo (ver Figura 3.4).

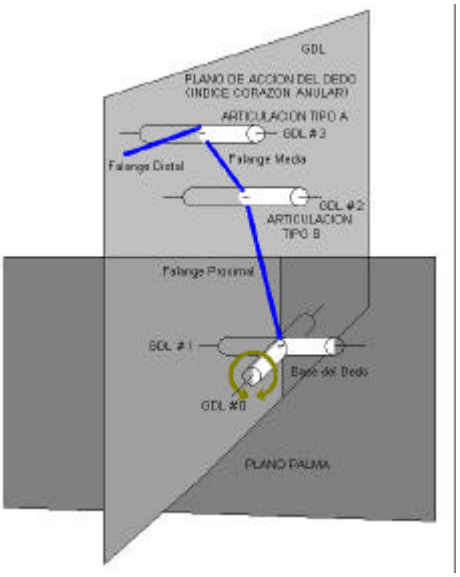


Figura 3.4 – Esquema de los GDL de un dedo. Fuente: IOC.

La construcción de estas articulaciones está basada en cuatro conjuntos de engranajes que redirigen el sentido de giro a 90º. Para diseñar esta articulación se partió de la premisa de que los motores que moverían los dos GDL deberían alojarse y fijarse sobre la palma [Suárez y Grosch, 2003, p.7]. El mecanismo adoptado permite el diseño deseado, pero obliga a trabajar a los dos motores simultáneamente para el movimiento de un GDL.

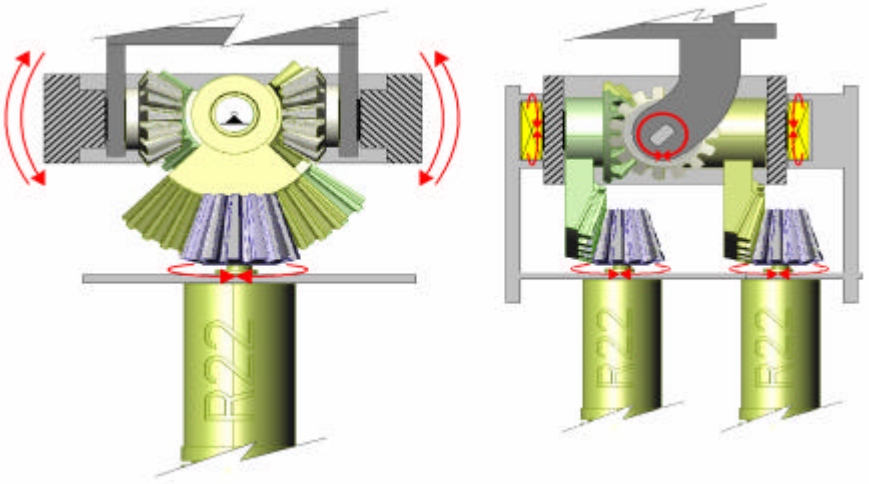


Figura 3.5 – Articulación de la falange proximal con la palma de la mano. Fuente: IOC

Para mover el dedo según el GDL #0 se deben hacer girar en el mismo sentido y el mismo ángulo los dos motores. De forma análoga para el GDL #1, los motores deben girar el mismo ángulo, pero en sentido contrario. Si se desea realizar un movimiento compuesto (ángulo AA para la abducción y AF para la flexión) basta que los motores giren un ángulo AM0 y AM1 según las ecuaciones (Ec. 3.1) y (Ec. 3.2).

$$AL = \left(\frac{AM0 + AM1}{2} \right) \cdot \frac{13}{49} \qquad \text{(Ec. 3.1)}$$

$$AF = \left(\frac{AM0 - AM1}{2} \right) \cdot \frac{13}{49} \cdot \frac{27}{15} \qquad \text{(Ec. 3.2)}$$

3.1.2. Hardware de Control

El sistema que controla los movimientos de la mano está formado por los siguientes componentes:

- Ordenador.

- Cuatro tarjetas I/O
- Etapa de potencia
- Conjuntos motor-reductor-codificador
- Sensores de final de carrera.

El ordenador se utiliza como controlador digital y de interfase con el usuario. Para ello, consta de cuatro tarjetas de entrada y salida instaladas en los puertos PCI del mismo, donde se reciben las señales de posición de los codificadores y envía las señales de consigna a la etapa de potencia. El sistema operativo instalado es QNX, una plataforma de tiempo real.

Las tarjetas I/O son de la firma Precision Micro Dynamics Inc (PMDI) y modelo MFIO-4A I/O preparadas para trabajar sobre plataformas de tiempo real. Cada tarjeta está asignada a un dedo, muestreando las señales que recibe de los cuatro codificadores y procesando las ocho señales digitales de los finales de carrera (dos por cada articulación). También es función de las tarjetas el envío de las cuatro señales de control (una para cada motor) a la etapa de potencia, utilizando como salida un potencial eléctrico de ±10V que indica la corriente que debe fluir por el motor correspondiente.

La etapa de potencia está completamente diseñada y construida en el IOC. Sus funciones son: transformar la consigna recibida de las tarjetas en la corriente equivalente de alimentación para cada motor, limitar la corriente que llega a los motores al rango de trabajo de éstos y concentrar todas las señales, que van o vienen de la mano, en ocho cables planos de 60 hilos que se conectan a las tarjetas de adquisición instaladas en el ordenador. Finalmente cabe señalar que el sistema tiene un consumo máximo de 3 amperios por dedo cuando todos los motores giran a su régimen máximo [Suárez y Grosch, 2003, p.9].

Los conjuntos motor-reductor-codificador están formados por motores de corriente continua, un grupo reductor interno al motor, y un codificador incremental instalado en el eje del motor. La salida del codificador corresponde a la posición del eje motor, lo cual implica que la posición de la articulación se mide indirectamente como función de la relación del reductor y de los engranajes de la articulación.

Los finales de carrera se utilizan tanto para calibrar los valores de posición de cada articulación como para proteger los motores y los engranajes. Para realizar la calibración, los motores giran en un sentido determinado hasta que llegan a activar los finales de carrera, en ese instante se colocan los valores iniciales de posición en los respectivos registros y sirven de referencia para los movimientos posteriores. Tal y como se ha

mencionado anteriormente, los finales de carrera también se utilizan para proteger la mecánica de la mano, de tal modo que cuando son activados durante algún movimiento, el software de la mano corta la corriente de los motores responsables de la colisión.

El esquema del hardware de control se ilustra en la Figura 3.6.

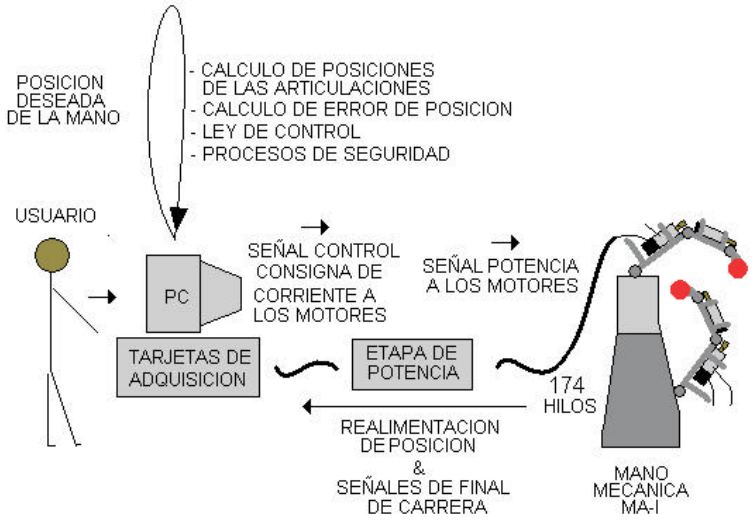


Figura 3.6 – Esquema del hardware de control. Fuente: IOC.

3.1.3. Software de la mano

El software desarrollado por el IOC para la mano (llamado MAIQNX6) se ejecuta sobre el sistema operativo de tiempo real QNX6 con el propósito de poder cumplir las especificaciones de control de la mano. En otros términos, el sistema operativo y el software deben garantizar que el período de muestreo de 1ms que se ha establecido para el controlador se cumple en todos los casos [Langsch, 2005, p.36].

MAIQNX6 está formado por cuatro módulos:

- Interfase
- Pathplanner
- Gestor de consignas
- Controlador

La interfase permite el movimiento de la mano según la consigna del usuario. El MAIQNXG permite introducir la posición deseada de todas las articulaciones, realizar la calibración

de la mano, ejecutar movimientos incrementales de articulaciones, etc. El sistema de ventanas de la interfase está programado sobre Photon, que proporciona un entorno como el que se puede observar en la Figura 3.7.

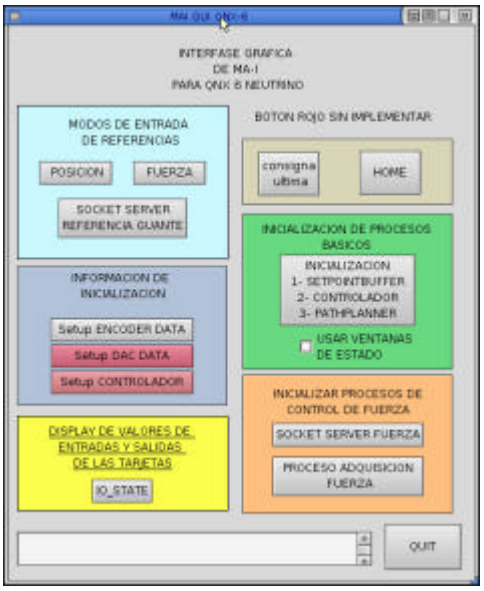


Figura 3.7 – Interfase MAIQNX6.

El módulo Pathplanner se ocupa de diseñar el perfil de movimiento que debe seguir cada articulación, con el propósito de que todas las articulaciones inicien y finalicen su movimiento simultáneamente. Los perfiles generados corresponden a una trayectoria descrita por una ecuación polinomial normalizada, ajustada por los parámetros posición inicial, posición final, velocidad de crucero y aceleración máxima permitida. Con estos datos se calcula un paquete de posiciones intermedias que cumple las restricciones para todos los dedos y se envía al Gestor de consignas.

El Gestor de consignas es básicamente una lista FIFO (lista en la que el primero en entrar es el primero en salir) de consignas de posición. Esta lista se llena según las demandas del Pathplanner y se vacía con cada período del controlador, ya que éste toma los valores de la lista para realizar el control.

El módulo Controlador es el que se ocupa directamente del control de la posición de las articulaciones de MA-I. Sus funciones son calcular la posición de la mano con los valores obtenidos de los codificadores, testar los finales de carrera y leer las consignas del Gestor de Consignas para calcular la tensión que será enviada a la etapa de potencia.

El proceso de inicialización que debe seguir el usuario para que MAIQNX6 funcione correctamente es el siguiente:

1. En primer lugar se enciende la etapa de poténcia.
2. Posteriormente se ejecuta el programa MAIQNX6 desde un terminal.
3. A continuación se debe pulsar el botón de Inicialización (este botón pone en marcha los procesos Pathplanner, Gestor de consignas y Controlador)
4. Finalmente, se pulsa el botón HOME para calibrar los codificadores incrementales. Este último paso es imprescindible, ya que si no se lleva a cabo, el control puede realizarse sobre unas lecturas de posición erróneas.

3.2. Brazo Robot

El brazo robot del IOC (ver Figura 3.8) pertenece a la firma Stäubli y corresponde con el modelo comercial TX-90. El armario de control que lo comanda es el CS8C, suministrado junto con la unidad de mando manual (MCP). Las aplicaciones del brazo robot están escritas en el lenguaje VAL3 y suelen ser diseñadas en el entorno para el desarrollo de aplicaciones SRS (Stäubli Robotics Studio).

Figura 3.8 – Brazo robot Stäubli TX-90. Fuente: [STÄUBLI, 2004b]

3.2.1. Brazo (TX-90)

El brazo robot del IOC corresponde al modelo TX-90. Las siglas TX indican que se trata de un robot de seis GDL con una baja capacidad de carga, mientras que el número 90 corresponde a las características geométricas del brazo (ver Figura 3.9), que se pueden consultar con mayor detalle en [STÄUBLI, 2004b, p. 24].

El brazo robot está constituido por miembros rígidos enlazados entre sí por articulaciones. Estos miembros se conocen como el pie (A), el hombro (B), el brazo (C), el codo (D), el antebrazo (E) y el puño (F). El pie sirve de base para el brazo y contiene las fijaciones para anclarlo a la bancada y diversas conexiones que se describen más adelante. El resto de miembros se denominan realizando un paralelismo con las partes de un brazo humano, pese a que su morfología y proporciones no corresponden con las del robot (ver Figura 3.9).

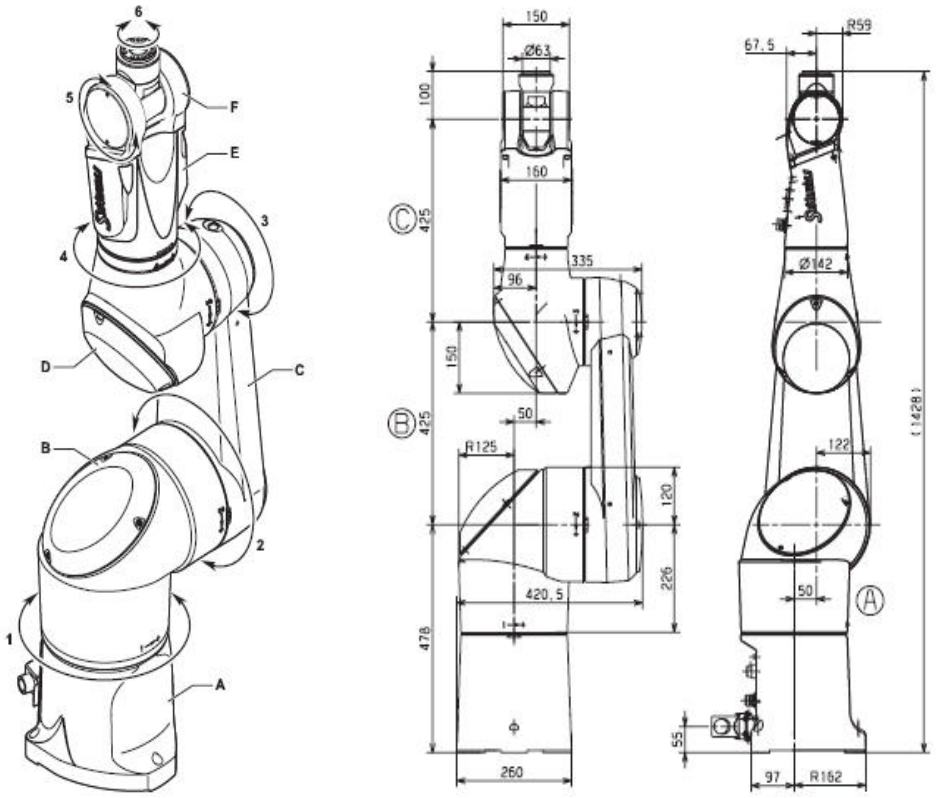


Figura 3.9 – Nomenclatura y geometría robot TX-90. Fuente: [STÄUBLI, 2004b].

A cada articulación del brazo se la conoce por un número, que va del 1 al 6, tal y como se muestra en la figura anterior. Los movimientos de cada articulación son generados por servomotores acoplados a sensores de posición. Cada uno de estos servomotores está

equipado con un freno de parada y de codificadores que permiten conocer la posición de la articulación. Además, cada articulación cuenta con dispositivos de seguridad de final de carrera.

La fijación del brazo a la bancada puede ser vertical (anclado en el suelo o el techo), u horizontal (anclado a un muro). El brazo robot del IOC se encuentra en posición vertical, anclado mediante tres tornillos M12 a la mesa metálica que sujeta los dos brazos de que dispone el laboratorio. Cabe destacar que el soporte debe ser lo más rígido posible para no reducir las prestaciones de velocidad y precisión del robot, siendo capaz de soportar en todo momento esfuerzos verticales de 1767N, esfuerzos horizontales de 1144N, y momentos de 1350Nm y 735Nm, en el eje horizontal y vertical respectivamente.

El robot TX-90 es capaz de operar en ambientes con una temperatura de entre 5 y 40 grados centígrados, una humedad relativa entre el 30 y el 95 por ciento (sin condensación), y una altitud máxima de 2000m. Su peso aproximado es de 111kg y su repetibilidad (a temperatura constante) es de ±0.03mm.

En la Tabla 3.2, que se muestra a continuación, se indica la amplitud, velocidad y resolución de cada articulación del brazo TX-90.

Eje	1	2	3	4	5	6
Amplitud [º]	360	277.5	280	540	255	540
Distribución de amplitud [º/s]	±180	+147.5 -130	±145	±270	+140 -115	±270
Velocidad Nominal [º/s]	250	200	300	430	350	600
Velocidad Máxima [º/s]	400	400	400	500	450	720
Resolución [º $\cdot 10^{-3}$]	0.057	0.057	0.057	0.057	0.122	0.183

Tabla 3.2 – Valores de amplitud, velocidad y resolución por eje.

El robot del IOC tiene integrado en su estructura un circuito neumático y eléctrico que pueden ser utilizados por el usuario mediante las conexiones de salida alojadas en el antebrazo del robot (toma de usuario). Actualmente no se utilizan estas conexiones para alimentar la mano.

El brazo TX-90 se alimenta mediante una manguera conformada por varios cables que se destinan a los servomotores, las electroválvulas, los finales de carrera y la toma de usuario. La conexión se realiza en el pie del brazo (ver Figura 3.10) donde hay los siguientes componentes:

- Toma de interconexión brazo/armario de mando (1).
- Toma eléctrica de usuario de tipo R23 (J1201).
- Selector de liberación de frenos (2).
- Pulsador de liberación de frenos (3).
- Conexión neumáticas a la red de aire comprimido P1 y P2.
- Silenciador de escape neumático (flecha)
- Borne de toma a tierra del brazo (4).

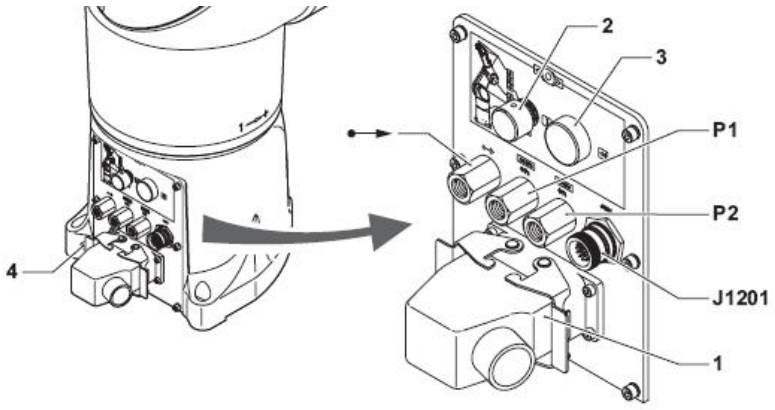


Figura 3.10 – Conexiones del brazo TX-90. Fuente: [STÄUBLI, 2004b].

El puño del robot contiene la interfase mecánica que permite la fijación de una herramienta que en este proyecto es la mano MA-I. Dicha herramienta se acopla a la interfase mecánica del robot mediante 4 tornillos M6, ajustados por un par de apriete 16.7Nm ±

1.2Nm. El posicionamiento relativo de la herramienta al puño se realiza mediante un pasador de diámetro 6 (ver Figura 3.11).

Nótese que la posición de la herramienta respecto al puño, una vez posicionada por el pasador, está desplazada 22º respecto a la posición de origen de la articulación 6. Esto debe tenerse en cuenta si el adaptador de la herramienta que se va a utilizar no corrige este desplazamiento (como es el caso de MA-I).

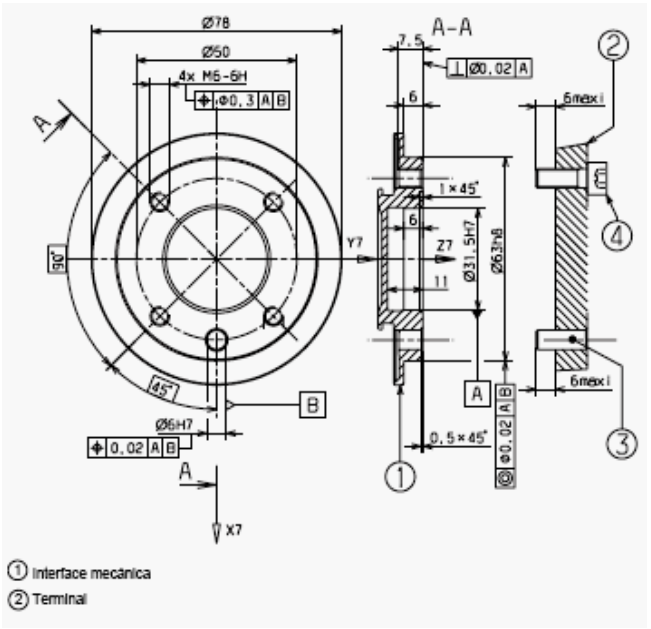


Figura 3.11 – Interfase robot TX-90 / herramienta. Fuente: [STÄUBLI, 2004b].

3.2.2. Armario de Control (CS8C)

El brazo robot del IOC está controlado y alimentado por el CS8C (ver Figura 3.12 del armario de control), que consta de los siguientes elementos:

- Los amplificadores de potencia numéricos (1) que alimentan cada eje del brazo.
- La alimentación RPS (2) que suministra la energía a TX-90.
- La alimentación ARPS (3) que proporciona la tensión necesaria a la unidad de mando manual (MCP).
- La tarjeta RSI (4) que realiza las funciones de seguridad eléctrica.
- El calculador (5) que realiza el control para guiar al robot.

- El interruptor general (6) que permite el accionamiento y el paro del armario de control.

- El bloque de potencia PSM (7) que realiza la conversión de energía eléctrica que proviene de la red.

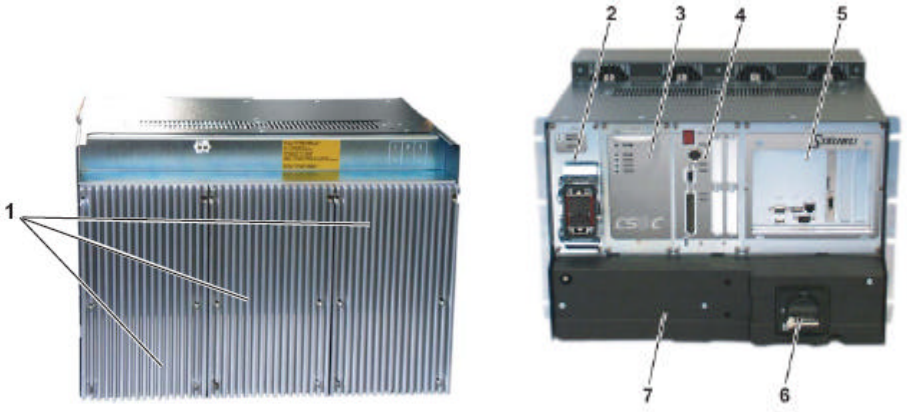


Figura 3.12 – Componentes del armario de control CS8C. Fuente: [STÄUBLI, 2004a].

El peso del armario de control es aproximadamente de 70kg y sus dimensiones máximas son 33.1cm de altura, 48.1cm de ancho y 45.4cm de profundidad.

La conexión del armario a la red eléctrica se hace a través de una conexión trifásica con toma a tierra. El CS8C está equipado con un filtro (para limitar las perturbaciones emitidas por el armario de control) que tiene su salida en la toma a tierra. La potencia instalada para la alimentación del CS8C es de 2kw. El armario de control alimenta a su vez el brazo TX-90.

El armario CS8C es capaz de operar en ambientes similares a los del TX-90, con una temperatura comprendida entre 5 y 40 grados centígrados, una humedad relativa hasta el 90 por ciento (sin condensación), y una altitud máxima de 2000m. El aire del ambiente debe ser limpio (clase 10 000, Federal Standard 209E). Nótese que el ambiente del IOC cumple con las especificaciones ya que no se trata de una industria (entorno que afecta negativamente a la calidad del aire) y el laboratorio cuenta con un sistema de climatización.

La refrigeración del armario (ver Figura 3.13) se realiza mediante ventilación filtrada, lo que obliga a mantener 5cm de separación con al menos las tres caras que contienen los orificios de entrada (2, 3) y salida de aire (1, 4). Para responder a estas necesidades, el

CS8C se aloja en un bastidor formado por perfiles metálicos, y unas bases roscadas de goma, que permiten colocar el armario en posición horizontal y lo protegen de vibraciones.

Figura 3.13 – Refrigeración del armario CS8C. Fuente: [STÄUBLI, 2004a].

Las conexiones que permiten el intercambio de información del armario con otros sistemas se encuentran en el panel frontal como se puede observar en la Figura 3.14. A continuación se detallan las más utilizadas:

- Un conector para la MCP (1).
- Un conector para recibir información de la célula de trabajo (señales de parada de emergencia, puerta abierta, etc.) (3).
- Un enlace Ethernet (5).
- Un puerto Serie (6).
- Un enlace USB (7).

Figura 3.14 – Conexiones CS8C. Fuente: [STÄUBLI, 2004a].

Resulta de especial interés profundizar en las funcionalidades que aporta la conexión ethernet (5) al armario de control. Como se verá más adelante, la comunicación del software del proyecto con el robot se realiza a través de este enlace.

El enlace ethernet del CS8C puede ser configurado para comunicarse mediante sockets a través de la aplicación "Panel de Control" (Control panel > I/O / Socket). En esta aplicación se deben introducir los parámetros del socket que va a ser utilizado, que puede ser de tipo cliente o servidor. Nótese que el CS8C soporta hasta 40 sockets simultáneos.

Los parámetros de los sockets de tipo servidor, implementados por STÄUBLI, son los siguientes:

- El puerto de conexión comprendido entre 1000 y 65535.
- La cantidad máxima de clientes simultáneos, valor por defecto 1.
- El plazo previo a activación de error (tiempo máximo de espera en lectura o en conexión). Un valor nulo elimina el control de tiempo de espera. Parámetro específico de STÄUBLI.
- El carácter de fin de cadena. Por el cual se determina que la recepción o el envío de un mensaje ha terminado. Parámetro específico de STÄUBLI.

Cabe destacar que los sockets de tipo servidor se activan en el CS8C en cuanto los utiliza un programa VAL3, y son desactivados cuando se desconecta el último cliente. En el caso de que se alcance la cantidad máxima de clientes que se han establecido para un socket servidor, se acepta a los otros clientes que traten de conectarse, pero la comunicación es inmediatamente interrumpida por el servidor.

El enlace ethernet también se utiliza para cargar los programas desarrollados mediante las utilidades descritas en el apartado 3.2.4 en el armario CS8C. Para lo cual basta con definir la dirección IP del armario de control, y acceder a él según el protocolo FTP (Siglas del protocolo de transferencia de archivos en Internet: File Transfer Protocol). Nótese que el usuario y contraseña de red necesarios corresponden al perfil de usuario definido en el CS8C y que el derecho de acceso en lectura y escritura depende del perfil de usuario seleccionado.

3.2.3. Unidad de mando manual (MCP)

La MCP (ver Figura 3.15) permite al usuario operador aplicar la potencia, seleccionar los modos de funcionamiento, administrar las aplicaciones, accionar elementos de seguridad, regular la velocidad y controlar los movimientos del robot.

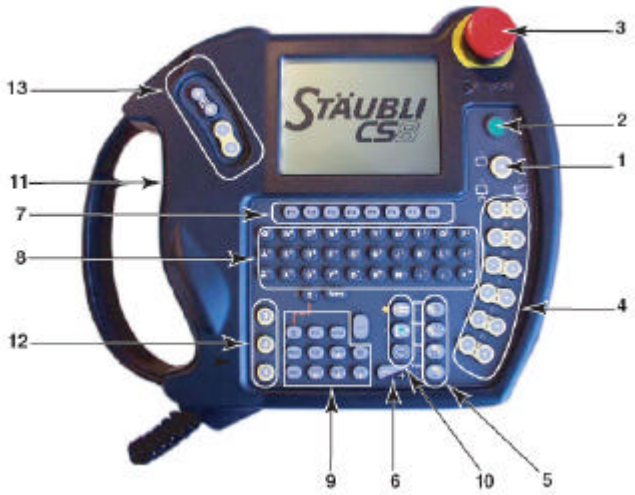


Figura 3.15 – Unidad de mando manual MCP. Fuente: [STÄUBLI, 2004a].

A continuación se describen las funciones de los botones más utilizados de la MCP:

- Botón de selección del modo de marcha (1). Este pulsador permite seleccionar uno de los cuatro modos de marcha (modo prueba, modo manual, modo local y modo desplazado). El botón se encuentra rodeado de los pictogramas de los cuatro modos, permaneciendo siempre iluminado el modo activo mediante un LED (diodo emisor de luz).

- Botón de aplicación de potencia al brazo (2). Este botón permite aplicar o cortar la potencia al brazo. Si el botón está iluminado, la aplicación de potencia es efectiva.

- Botón de validación (corresponde a la marca #11, situado en el anverso de la MCP). Este botón tiene tres posiciones: activado, desactivado y Crispación del usuario (cuando se presiona con fuerza). Este botón autoriza la aplicación de potencia al brazo en modo manual únicamente en la posición de activado.

- Parada de emergencia (3). La parada de emergencia desactiva la potencia del brazo y lo frena. Este botón sólo debe utilizarse en caso de necesidad absoluta ya que su uso continuado disminuye el tiempo de vida de los componentes del brazo.

- Teclas de movimientos (4). Estas teclas permiten (en el modo de marcha manual) generar movimientos del brazo según el modo de desplazamiento elegido.

- Teclas de elección del modo de desplazamiento (5). Estas teclas permiten seleccionar el modo de desplazamiento elegido (Joint, Frame, Tool o Point)

cuando la potencia está habilitada y el modo de marcha es manual. La tecla que indica el modo de desplazamiento en curso permanece iluminada.

- Tecla reguladora de la velocidad (6). Esta tecla permite introducir la velocidad con la que se mueve el robot, dentro de los límites de cada modo de desplazamiento, obtenida como el porcentaje de la velocidad máxima aplicable en cada caso. El valor de la velocidad se muestra en todo momento en la barra de estatus de la pantalla de la MCP (punto 5 de la Figura 3.16).

- Teclas de menús contextuales (7). Se utilizan para interactuar con los menús contextuales mostrados encima de ellas.

- Teclas de interfaz y de navegación (9). Estas teclas permiten la navegación del usuario por la interfaz gráfica de la MCP.

- Teclas de comando de las aplicaciones (10). Este grupo está formado por tres teclas: Stop, Run y Move/Hold. La tecla Run inicia una aplicación VAL3 de las que se hayan abierto en la TCP mediante las teclas de interfaz y de navegación. La tecla Stop detiene la aplicación que se está ejecutando. La tecla Move/Hold ejerce las funciones de pausa en los modos manual, local y desplazado, mientras que en el modo prueba requiere de su activación para el movimiento del brazo.

La pantalla de la CMP se subdivide en las zonas A, B y C (ver Figura 3.16) que se discuten a continuación:

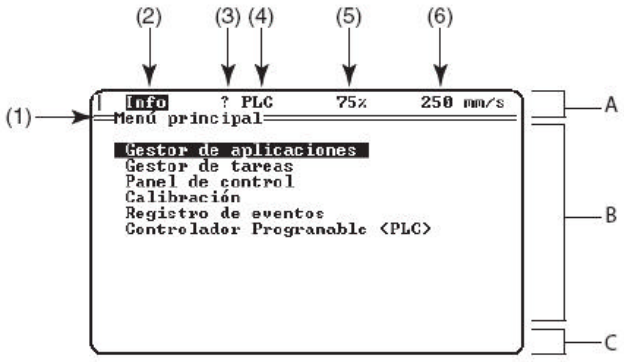


Figura 3.16 – Pantalla de la MCP. Fuente: [STÄUBLI, 2004a].

La barra de estado (A) muestra mensajes de tipo informativo como el indicador de presencia de nuevos mensajes de información (2) o el indicador de velocidad de desplazamiento del brazo (5). En el caso de que haya algún mensaje de información almacenado en el historial de acontecimientos el indicador de presencia de nuevos

mensajes de información parpadea hasta su consulta por parte del usuario. El indicador de velocidad de desplazamiento del brazo muestra en todo momento la velocidad programada para el giro de las articulaciones del robot.

La página de trabajo (B) es la zona de la pantalla en la que se muestran las ventanas de información, las paginas de introducción de datos y se muestra toda la información relativa a la aplicación que se está ejecutando.

Los menús contextuales (C) permiten implementar teclas virtuales en función de la página de navegación, del elemento seleccionado o del contexto en general. Para pulsar la tecla virtual deseada basta con pulsar la tecla de menú contextual situada debajo.

3.2.4. Entorno para el desarrollo de aplicaciones VAL3 (SRS)

El entorno para el desarrollo de aplicaciones VAL3 se conoce como SRS y está formado por un conjunto de aplicaciones que reúnen todas las herramientas disponibles para desarrollar y mantener una aplicación en un robot Stäubli.

Las aplicaciones del SRS que se han utilizado para el desarrollo de este proyecto se describen a continuación:

- VAL3 Studio: Editor de lenguaje VAL3, que permite verificar la sintaxis del programa en desarrollo. El editor también incluye la herramienta Downloading Tool que permite enviar y recibir fácilmente una aplicación VAL3 o ficheros de entrada/salida, a un CS8C.

- Emulador CS8C. Es un emulador completo del armario de control CS8C, sobre el cual se pueden ejecutar los programas en desarrollo.

- 3D Studio. Aplicación que muestra un modelo virtual del robot TX-90 que se mueve según las consignas del armario de control al que esté conectado.

Nótese que un método interesante para el testado de aplicaciones, antes de su ejecución real, es editar la aplicación utilizando VAL3 Studio (verificando la sintaxis), ejecutar el programa sobre el Emulador CS8C, y ver los movimientos del brazo mediante el 3D Studio, previa conexión al Emulador.

3.2.5. Célula de trabajo y protocolos de actuación

La seguridad en el entorno de trabajo de los robots industriales es un aspecto que debe tenerse en cuenta, ya que estas máquinas pueden provocar lesiones muy graves sobre las personas en caso de accidente. El laboratorio de robótica del IOC está equipado con

una célula de trabajo, donde se encuentran los dos robots de que dispone el instituto, y ha establecido una serie de protocolos de actuación como medidas de seguridad. Al mismo tiempo, el laboratorio del IOC está dotado con diversos circuitos de parada de emergencia.

La célula de trabajo consiste en una sala separada del resto del laboratorio por una estructura de aluminio y plástico transparente. Esta estructura contiene una puerta de acceso dotada de un mecanismo de seguridad.

Figura 3.17 – Célula de trabajo.

Los circuitos de parada de emergencia que están instalados en el laboratorio del IOC son los siguientes:

- El circuito integrado de emergencia de la MCP la activación del cual se produce al pulsar el botón (3) de la Figura 3.15.
- Cuatro circuitos en paralelo formadas por cuatro interruptores (como el mostrado en la Figura 3.18) distribuidos por las mesas del laboratorio.

Figura 3.18 – Interruptores de las cadenas de paro de emergencia.

- Una cadena de seguridad formada por la puerta cuando está cerrada y en estado de seguridad. El estado de seguridad se activa al pulsar el botón azul que contiene la caja de seguridad del interior de la célula de trabajo, (1 de la Figura 3.19) y veinte segundos después (como máximo) cerrar la puerta y pulsar el botón situado en el exterior de la célula (2).

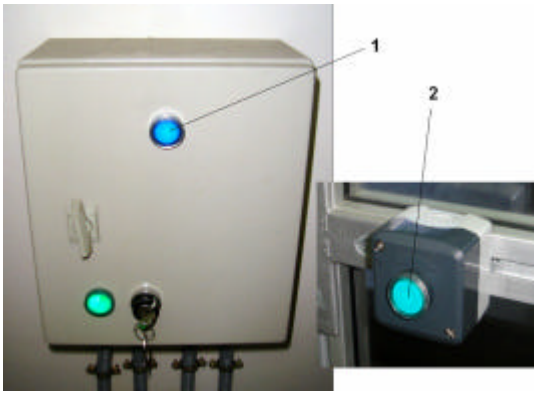


Figura 3.19 – Elementos de seguridad de la puerta.

Los protocolos de actuación utilizados durante la elaboración de este proyecto son la puesta en marcha y apagado del robot, el movimiento del brazo en modo manual y la ejecución de un programa en modo local.

Para realizar la puesta en marcha y el paro del robot se deben realizar los siguientes pasos:

1. Despejar el área de trabajo
2. Encender el controlador mediante el botón (6) de la Figura 3.12.
3. Observar que todos los LED's de la MCP parpadean, ya que si alguno de ellos no se enciende la MCP deberá ser cambiada.
4. Esperar a que el sistema se cargue correctamente. El controlador realizará dos pitidos y la MCP mostrará el menú principal en su pantalla.
5. Realizar el trabajo.

6. Una vez finalizado el uso del robot, se detendrá el brazo y se cortará la potencia del TX-90 (mediante la tecla 2 de la Figura 3.15)

7. Finalmente se apagará el sistema (desactivando el interruptor general, botón 6 de la Figura 3.12.).

El protocolo de actuación para realizar movimientos del brazo en modo manual se describe a continuación:

1. Ejecutar el protocolo de puesta en marcha del robot.

2. Seleccionar el modo de marcha manual mediante la tecla (1) de la Figura 3.15.

3. Habilitar la potencia del brazo pulsando el botón (2) de la MCP que se muestra en la Figura 3.15.

4. Seleccionar el modo de desplazamiento deseado (Joint, Frame, Tool, Point) pulsando la tecla adecuada del grupo de teclas (5) de la Figura 3.15.

5. Fijar la velocidad de movimiento a través de la tecla reguladora de velocidad (6) de la Figura 3.15 y observar que el valor es correcto en el indicador de velocidad de desplazamiento del brazo (ver indicador 5 en la Figura 3.16).

6. Realizar el movimiento deseado con las teclas de movimientos que se observan en la marca (4) de la Figura 3.15.

El protocolo para la puesta en marcha de una aplicación en modo local es el siguiente:

1. Ejecutar el protocolo de puesta en marcha del robot.

2. Abrir la aplicación deseada mediante la utilidad “Gestor de aplicaciones” del menú del MCP.

3. Seleccionar el modo de marcha local mediante la tecla (1) de la Figura 3.15.

4. Habilitar la potencia del brazo pulsando el botón (2) de la MCP que se muestra en la Figura 3.15.

5. Fijar la velocidad de movimiento a través de la tecla reguladora de velocidad (6) de la Figura 3.15 y observar que el valor es correcto en el indicador de velocidad de desplazamiento del brazo (ver indicador 5 en la Figura 3.16).

6. Ejecutar la aplicación abierta pulsando la tecla Run que se encuentra entre las teclas de comando de las aplicaciones (10) Figura 3.15. .

7. Pulsar la tecla Move / Hold, del grupo de teclas de comando de aplicaciones (10) Figura 3.15. La tecla deja el estado intermitente y el LED permanece encendido. A partir de ese instante, el robot puede moverse según las ordenes de la aplicación en ejecución.

4. Diseño e implantación del sistema VMR

En este capítulo se describe la fase de diseño e implantación del sistema VMR (siglas del sistema Visión Mano Robot). En primer lugar se enuncian las características que debe tener el sistema. A continuación se describen los recursos con los que se va a dotar el laboratorio de robótica del IOC para cumplir con los requerimientos. Finalmente, se describe el software diseñado para el desarrollo de tareas de localización, aprehensión y manipulación de objetos.

4.1. Requerimientos

Previamente al inicio de este proyecto, los investigadores del IOC plantearon una serie de funcionalidades que debe aportar el sistema de visión al sistema MR. La función principal del nuevo sistema es proporcionar una plataforma de pruebas experimentales para los diferentes algoritmos y estrategias de localización, aprehensión y manipulación de objetos que se desarrollan en el IOC.

A continuación se describen con más precisión los requerimientos del sistema VMR:

- El software resultante del proyecto debe ser multiplataforma. En concreto, el software debe ser compatible con los sistemas operativos Windows XP y Linux.

- El sistema de visión deberá ser capaz de localizar objetos estáticos en 2D, integrando las herramientas necesarias para este fin. Al mismo tiempo, el sistema no debe contener limitaciones que supongan un problema para una futura integración de visión en 3D.

- La interfase desarrollada deberá ser capaz de enviar consignas de movimiento a la mano y el robot remotamente. También debe poder realizar tareas de localización sobre imágenes que se hayan tomado mediante otro sistema. Esta funcionalidad permitirá que la interfase pueda ejecutarse desde cualquier PC del laboratorio.

4.2. Recursos

Una vez fijados los requerimientos, se decidió que los recursos con los que se equiparía el sistema VMR serían una tarjeta de adquisición de vídeo, un PC y una cámara de vídeo en color. En los siguientes subapartados se detallan las características que justifican su elección, así como la instalación en el laboratorio del IOC.

4.2.1. PC (COBE)

El laboratorio del IOC está equipado para realizar sus actividades de investigación experimental con la MA-I con dos ordenadores. El ordenador llamado EL-FEO realiza el control de la mano (sistema operativo QNX6) y otro PC, cuyo nombre es COBE, contiene las aplicaciones que controlan el sistema háptico (una interfaz que interacciona con el ser humano mediante el sentido del tacto) de la mano.

Dado que las tareas de control de la mano que realiza El-Feo son de alta prioridad y en tiempo real, se decidió que no era necesario sobrecargar este PC con el sistema VMR. Sin embargo, COBE tiene capacidad para contener el VMR y el sistema háptico a la vez. En definitiva, se decidió utilizar COBE para el desarrollo del proyecto, evitando así la disgregación de las aplicaciones relacionadas con la mano.

Las características principales de COBE son las siguientes: Sistema operativo Microsoft Windows XP, procesador Intel modelo Pentium 4 a 2.4GHz, 512 MB de memoria RAM y 27.9GB de disco duro.

4.2.2. Tarjeta de adquisición de vídeo (Angelo RTV-24)

La tarjeta de adquisición de imágenes que utiliza el sistema VMR es una Angelo RTV-24 de la firma ADLINK Technology Inc (ver Figura 4.1).

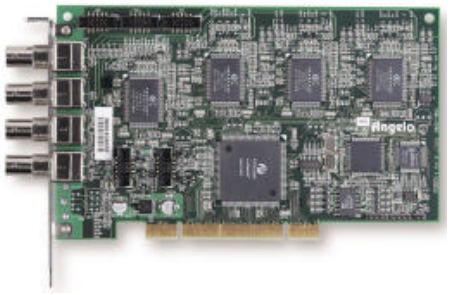


Figura 4.1 – Tarjeta de adquisición de imágenes Angelo RTV-24.

La tarjeta Angelo RTV-24 se puede utilizar bajo los sistemas operativos Windows y Linux. El fabricante proporciona un controlador de la tarjeta (apto para Windows) que puede utilizarse para programar aplicaciones en los siguientes lenguajes: C++ (lenguaje desarrollado en los 80 que abarca la programación estructurada, la programación genérica y la programación orientada a objetos) Visual Basic y Delphi. En caso de la instalación de la tarjeta en una máquina Linux se puede recurrir a la biblioteca Video for Linux que es compatible con la tarjeta y permite programar aplicaciones en C++.

La tarjeta de adquisición puede recibir la señal de vídeo a través de cualquiera de los cuatro conectores de tipo BNC que contiene. La señal de vídeo puede ser de tipo PAL/NTSC (en color) o de tipo CCIR / EIA (en blanco y negro). En cualquiera de los casos la adquisición de imágenes se realiza a razón de 30 fotogramas por segundo en cada canal.

La Angelo RTV-24 está equipada con cuatro entradas y cuatro salidas digitales de tipo TTL, y cuatro salidas de pulsaciones (software trigger outputs) protegidas frente a descargas electroestáticas. La conexión de la tarjeta con el PC se realiza en un bus de 32-bits, 33MHz/66MHz PCI.

El proveedor también suministra la utilidad View Creator que puede ser útil para evaluar el funcionamiento inicial de la tarjeta así como para el desarrollo de nuevas aplicaciones.

El controlador proporcionado por ADLINK permite (entre otras funciones) detectar y ajustar el formato del video que recibe la tarjeta, recortar una zona de interés en la imagen y/o escalarla antes de ser transferida a la memoria del PC, elegir el formato de color de la imagen (entre RGB, YUV, y el blanco y negro), ajustar la tonalidad, el brillo, el contraste y la saturación de la imagen y guardar un fotograma en un archivo de tipo TIF, BMP, o JPEG.

4.2.3. Cámara de vídeo (FCB-IX45CP)

La cámara de vídeo escogida para la captura de imágenes, es una FCB-IX45CP, de la marca SONY (ver Figura 4.2).

Figura 4.2 – Cámara de vídeo FCB-IX45CP de SONY.

Las características que debe tener la cámara de video del sistema VMR son bastante comunes, debido a la flexibilidad de la tarjeta Angelo RTV-24. Por esta razón, se escogió este componente que ya pertenecía al laboratorio y no estaba en uso en la actualidad.

La cámara del sistema es un elemento que puede ser reemplazado con facilidad. En ese caso, la nueva cámara debe cumplir con las siguientes especificaciones: que proporcione una señal de tipo PAL/NTSC (estos códigos son en color y pueden ser útiles para poder realizar algoritmos de distinción de color), una resolución mínima de 384X288 píxeles (resolución total de 110592 píxeles), y de bajo consumo (por razones económicas y medioambientales).

Las principales características técnicas de la FCB-IX45CP se describen a continuación: alimentación 12V, peso de 170g, dimensiones 48.2x56.6x92.3mm, dieciocho aumentos ópticos y cuatro digitales, consumo de 1.5w con los motores parados y de 2w con los motores en movimiento, temperatura de operación entre 0 y 50ºC, conexión de salida de vídeo BNC, resolución de 440000 píxeles, luminancia mínima de un lux (aunque la luminancia recomendada está entre 100 y 100000lx), y funciones automáticas como la SMART(Sony Modular Automatic lens Reset Technology) o el autoenfoque.

4.2.4. Instalación hardware

La instalación de la tarjeta Angelo RTV-24 en COBE se realizó según las indicaciones del fabricante en el bus PCI de la placa base del PC. Una vez instalados los controladores de la tarjeta, se comprobó su buen funcionamiento mediante la aplicación View Creator.

Dado que el sistema de visión solo cuenta con una cámara, es necesario fijar una serie de parámetros para facilitar la localización de un punto en el entorno de trabajo (en este caso la mesa metálica que soporta los dos robots). Con este propósito la FCB-IX45CP está situada con una orientación y una posición fijas.

La orientación de la FCB-IX45CP es tal, que el plano que contiene la lente de la cámara es paralelo al plano de la mesa de trabajo. De este modo se puede realizar un escalado para establecer la relación entre los puntos de la imagen y los puntos de la mesa (si se desprecia la deformación de la lente). En la Figura 4.3 se puede observar el efecto que causa tomar imágenes con la cámara inclinada respecto a la vertical en la relación de los puntos de la imagen con los puntos de la mesa.

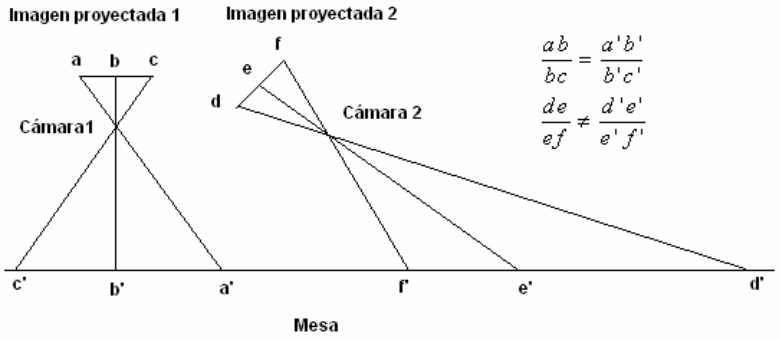


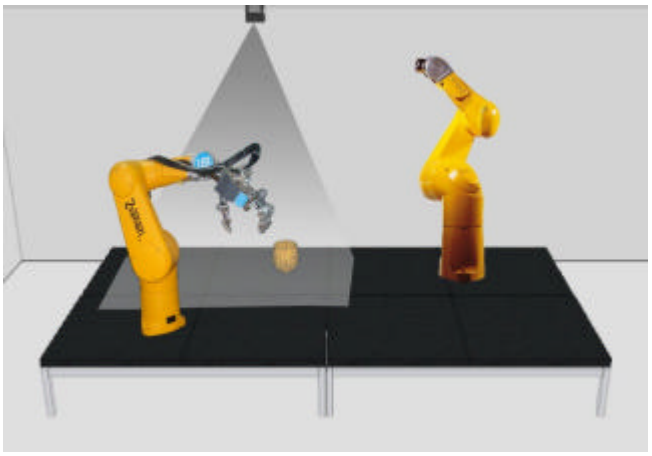
Figura 4.3 – Efecto de la inclinación de la cámara en las imágenes.

Para situar la cámara en una posición e inclinación fijas, se utiliza una estructura de aluminio instalada en el techo de la célula de trabajo. Esta estructura (que también alberga la fuente de alimentación de la cámara) está situada fuera del espacio de trabajo del sistema MR, evitando, por tanto, la colisión de los robots o de la mano con cualquier elemento que la conforma.

La sujeción de la cámara a esta estructura se realiza mediante una pletina de aluminio atornillada a su carcasa y a una barra (ver Figura 4.4). Esta pletina permite mover la cámara para obtener la orientación deseada, mientras que la barra define la posición que tiene respecto al robot.

Figura 4.4 – Instalación de la cámara en la estructura del laboratorio.

Finalmente se muestra una representación de la célula de trabajo, con el campo de visión aproximado de la cámara instalada (Figura 4.5).



4.3. Software del sistema VMR

Para abordar la descripción del software que ha sido desarrollado para el sistema VMR se divide su estudio en cinco subapartados. En primer lugar se describe el esquema funcional establecido en la fase de diseño. A continuación, se profundiza en las cuatro funciones básicas del software VMR: captura y tratamiento de imágenes, determinación de consignas, comunicación con el sistema MR, así como la interacción con el usuario. Finalmente se presenta una guía de usuario del software implementado.

4.3.1. Esquema funcional

Si el proyecto se asimila a una caja negra, se obtiene que la variable de entrada de la caja es la falta de una plataforma de pruebas experimentales de diferentes algoritmos de localización así como de diversas estrategias aprehensión y manipulación de objetos mediante sistema MR. La función de la caja negra es la capacitación del sistema MR para realizar este tipo de investigaciones.

Desde el punto de vista del software del sistema VMR, el modelo de caja negra tiene como variables de entrada el usuario, un objeto sobre la plataforma de trabajo, el sistema MR y el hardware de visión. La caja negra debe realizar las funciones necesarias para que su variable de salida sea un objeto localizado, aprehendido y manipulado. Las funciones necesarias para llevar a cabo este proceso se describen a continuación:

- El VMR ha de contener una función de interacción con el usuario, para que éste pueda comandar las acciones que realizará el sistema.

- El sistema también debe proporcionar una función de captura y tratamiento de imágenes, con tal de localizar el objeto que se encuentra sobre la plataforma de

trabajo. Nótese que esta función es flexible y permite, por tanto, el desarrollo de técnicas de localización aún no implementadas.

- La aprehensión y manipulación de objetos con el sistema MR se satisface mediante la función de determinación de consignas, que también es susceptible de ser ampliada.

- Finalmente, el sistema VMR contiene una función de comunicación con el sistema MR, que permite el comando de los movimientos del sistema MR.

Para dotar de mayor versatilidad al sistema, se ha diseñado el VMR de tal modo que el uso de las tres últimas funciones se puede realizar individualmente. En otros términos, se puede capturar una imagen para procesarla sin necesidad de aprehender un objeto, como también resulta factible, por ejemplo, el comando del sistema MR independientemente del sistema de visión.

A continuación se muestra un esquema funcional del software del sistema VMR, detallada también la relación entre las funciones y la conexión del software con las entradas del modelo de caja negra (ver Figura 4.6).

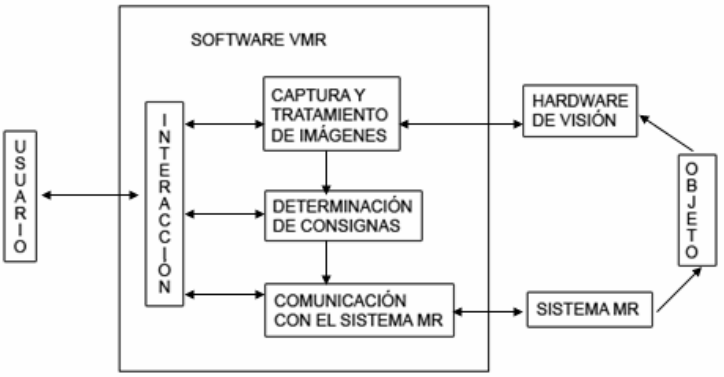


Figura 4.6 – Esquema funcional del software VMR.

4.3.2. Captura y tratamiento de imágenes

Con el propósito de capacitar al sistema VMR para el reconocimiento y localización de objetos en el entorno de trabajo, se ha desarrollado un software que realiza la función de captura y tratamiento de imágenes.

El software debe ser multiplataforma y estar capacitado para procesar imágenes que se hayan tomado desde otro sistema, para lo que se ha optado por separar la función de visión en dos. En primer lugar se ejecuta la captura de imágenes y a continuación se realiza su tratamiento.

La obtención de imágenes se efectúa mediante un programa independiente llamado *captura.exe* que captura un fotograma a través de la tarjeta Angelo RTV-24 y lo guarda como un archivo de mapa de bits llamado *imagen.bmp*.

La independencia de la aplicación respecto al software VMR permite que la imagen pueda ser capturada mediante otro dispositivo, adaptándose con facilidad a un cambio de la tarjeta de adquisición, a la instalación de la tarjeta en un sistema operativo diferente o incluso la toma de imágenes desde un PC remoto. En estos casos lo único que debe hacerse es implementar un programa llamado *captura.exe* que obtenga una imagen de la tarjeta de adquisición o de un PC remoto y guardarla con el nombre *imagen.bmp*. Para el correcto funcionamiento del software VMR conviene que este programa se guarde en el mismo directorio del ejecutable *vmr.exe*.

El programa *captura.exe* diseñado para este proyecto, se ha implementado en lenguaje C++, utilizando para su desarrollo el entorno Visual Studio. El ejecutable implementado es una aplicación de la línea de comandos de Windows.

Los archivos más importantes del proyecto Captura generado en Visual Studio se encuentran disponibles en el Anexo A. Nótese que la aplicación toma la imagen del puerto cero de la tarjeta Angelo RTV-24, y por lo tanto la señal de vídeo de la cámara tiene que conectarse en esa toma de la tarjeta.

La distinción y localización de objetos estáticos en 2D es posible gracias a la inclusión en el software VMR de la librería Computer Vision de la marca Intel. Esta biblioteca es multiplataforma, gratuita y de código abierto, por lo que sólo se debe preservar el copyright para utilizarla (la aceptación de la licencia está incluida en los archivos del software que la usan).

La librería de Intel contiene herramientas de manejo de archivos, matemáticas de matrices, operaciones lógicas de imágenes, filtrado, cambio del espacio de color, transformaciones geométricas de gráficos, procesado de contornos, estadística de la imagen, acceso rápido a píxeles, morfología, diferenciación de fondos, calibrado de cámaras, plantillas de movimiento, dibujo de formas y textos en la imagen, etc.

Las características de la librería descritas con anterioridad hacen de ésta una herramienta adecuada para el uso, la modificación, y la incorporación por parte de los investigadores a sus algoritmos de distinción y localización de objetos.

La descarga de la versión utilizada en el proyecto se realizó desde la página http://sourceforge.net/projects/opencvlibrary/ el 20 de Abril de 2007, link obtenido a través de la página oficial de Intel: http://www.intel.com/technology/computing/opencv/. La instalación de la librería se llevó a cabo según las instrucciones de Intel, disponibles en la dirección de Internet citada.

Para el buen funcionamiento de la librería, es sumamente importante importar los archivos de cabecera y los de librería a los directorios adecuados del compilador. En el caso de la interfase VMR el compilador utilizado es el GCC (Siglas en inglés de Colección de Compiladores GNU) siendo este software de código abierto y distribuido por la Fundación de Software Libre mediante la licencia GPL (siglas de Licencia Pública General en inglés). Descargado el 8 de marzo de 2007 de la página http://gcc.gnu.org/.

Por lo tanto, para utilizar la Computer Vision Library de Intel, se deben instalar los archivos de cabecera en la carpeta *include* y los archivos de librería en la carpeta *lib* del directorio de instalación del GCC.

4.3.3. Determinación de consignas

Para poder realizar tareas de aprehensión y manipulación, resulta necesario determinar las consignas a ejecutar por el sistema MR. Para facilitar esta labor, se han integrado en el software del sistema VMR los siguientes componentes:

- La librería matemática MT desarrollada en el IOC, y que utiliza la biblioteca de código abierto BOOST.
- La librería de la cinemática directa e inversa del robot (denominada TXRobot) desarrollada por el IOC, que utiliza las librerías anteriores.
- La librería de la cinemática directa de la mano (llamada MAI) diseñada en el marco de la realización de este proyecto, que también utiliza la MT.

La biblioteca matemática MT contiene herramientas de uso frecuente en la determinación de consignas de sistemas robotizados. La librería incluye, entre otros, contenedores matemáticos (vectores, matrices, puntos, etc), operadores geométricos (rotación, translación, transformación, etc.), y elementos geométricos (círculos, planos, puntos, cilindros, etc.).

Para la correcta instalación de las librerías MT y BOOST solo se deben copiar los archivos que las conforman en dos carpetas llamadas *mt* y *boost* respectivamente, y pegarlas en la carpeta *include* del directorio de instalación del GCC.

A continuación se describe el funcionamiento de las bibliotecas TXRobot y la MAI, así como los procedimientos a seguir para su correcta utilización.

TXRobot

La librería TXRobot se ha desarrollado en el IOC y es ampliamente utilizada por bs investigadores del instituto en los proyectos relacionados con los brazos robot. Esta herramienta permite calcular la cinemática directa e inversa del robot. Los archivos de la librería, en lenguaje C++, se pueden consultar en el Anexo B.

En el caso de la cinemática directa, se deben introducir las posiciones articulares de los seis ejes del robot para obtener la posición y la orientación de la muñeca en la referencia del robot.

El origen del sistema de referencia del robot se encuentra situado a 478mm del centro de la base del brazo (dentro del hombro del robot). Su orientación se define por el eje Z, que se encuentra alineado con el eje vertical y con valores positivos por encima de la base. El eje Y coincide con el eje de la articulación 2 del robot, cuando éste se encuentra en posición de HOME siendo los valores positivos los que se sitúan del origen hacia el brazo del robot también en dicha posición HOME. Finalmente, la base está definida por el eje X que se obtiene a partir del producto vectorial entre el eje Y y el Z.

La referencia de la muñeca se sitúa en el centro de la interfase mecánica del TX-90 y su orientación coincide con la referencia del robot en la posición HOME. La situación y la orientación de las referencias del robot (rojo) y de la muñeca (verde) se pueden observar a continuación en la Figura 4.7.

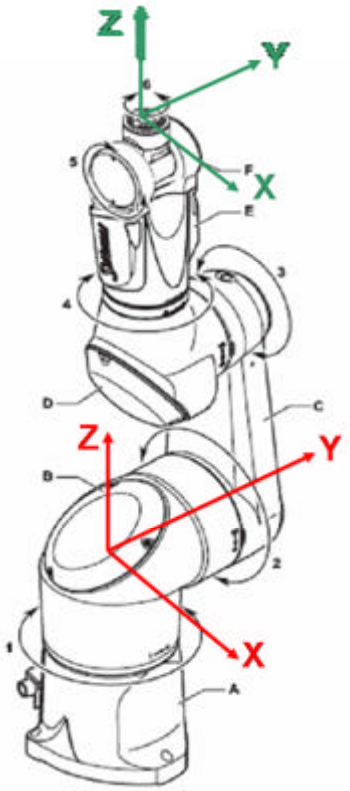


Figura 4.7 – Situación y orientación de las referencias robot y muñeca.

En el caso de la cinemática inversa, se parte de la posición y orientación de la muñeca en la referencia del robot y se obtienen las seis posiciones articulares de los ejes del robot. Cabe destacar que en general, el robot tiene varias posibilidades para alcanzar una posición cartesiana dada. Por esta razón, y para limitar las posibles soluciones a una sola, la biblioteca TXRobot fija la configuración con la que el brazo puede llegar al punto deseado.

Las posibles configuraciones del brazo TX-90 se obtienen a partir de las configuraciones del hombro (srighty o slefty), el codo (epositive o enegative) y la muñeca (wpositive o wnegative).

En primer lugar, se define la configuración del hombro ya que para alcanzar un punto cartesiano, el brazo del robot puede situarse a la derecha (srighty) o a la izquierda (slefty) del punto. Del mismo modo, existen dos posibilidades la configuración del codo, una en la que se encuentra por encima del hombro del robot (epositive) y otra en que está por debajo (enegative). Finalmente la muñeca se puede configurar para que su posición

angular sea positiva (wpositive) o negativa (wnegative). En la Figura 4.8 se pueden observar las posibilidades de configuración de las tres articulaciones.

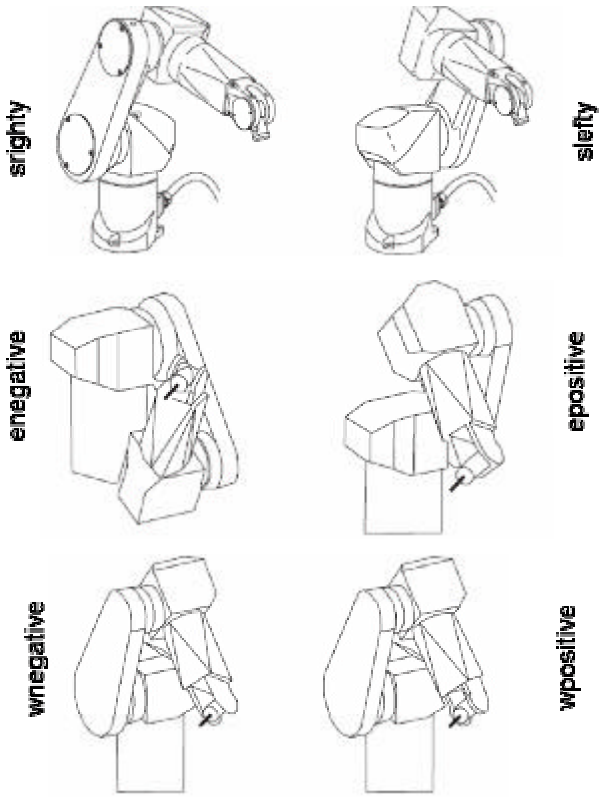


Figura 4.8 – Configuración del hombro, el codo y la muñeca del TX-90.

En resumen, existen 8 posibles configuraciones para el robot, debido a la combinatoria de las tres articulaciones. Nótese que en caso de que no se fije una configuración en la TXRobot, se aplicará la que hay por defecto en la resolución de la cinemática inversa (srighty, epositive, wpositive).

MAI

La librería MAI ha sido desarrollada, en el marco de este proyecto, en colaboración con el alumno de doctorado Carlos Rosales. El lenguaje de programación utilizado para la implementación de la librería es C++. La funcionalidad actual de MAI es calcular la cinemática directa de la mano, y se está estudiando la posibilidad de implementar también la cinemática inversa.

La biblioteca permanece a disposición del centro para ser utilizada y ampliada por los investigadores del instituto en los proyectos relacionados con manos mecánicas. Los archivos que la conforman se pueden consultar en el Anexo B.

Para calcular la cinemática directa de MA-I se ha modelado la mano mediante 4 cadenas cinemáticas según el método de Denavit-Hartenberg. Cada cadena comienza en la base de la mano y termina en el extremo de un dedo. La base (x0, y0, z0 de la Figura 4.9) está en el centro de la superficie inferior de la mano.

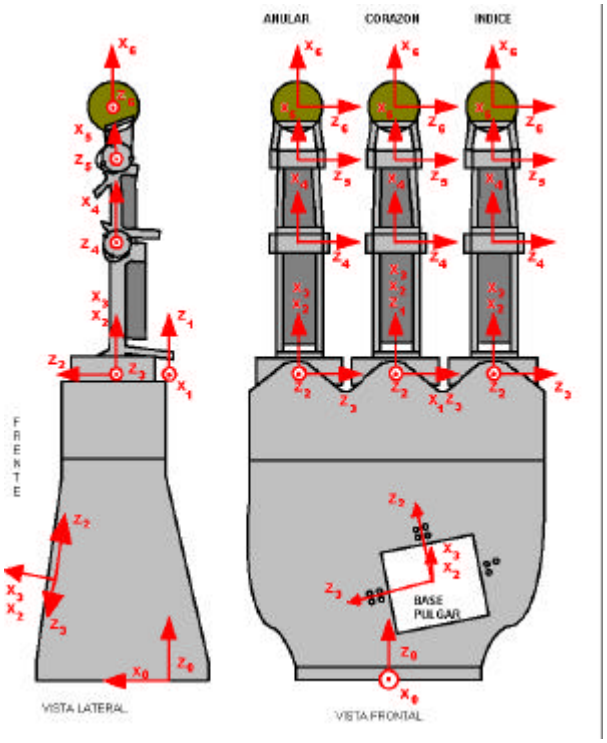


Figura 4.9 – Sistemas de referencia usados en el modelo DH de la mano. Fuente: IOC.

Después de la base se introduce una articulación ficticia 1 (x1, y1, z1 de la Figura 4.9) a lo largo del eje Z de ésta (la articulación ficticia 1 utilizada para el pulgar no se muestra en este dibujo para mayor claridad, pero se obtiene con los parámetros que se muestran más adelante), a continuación se disponen las referencias 2, 3, 4 y 5 que corresponden a los movimientos de las articulaciones de los dedos, y finalmente se llega a la referencia ficticia 6 que se sitúa en el centro de las yemas.

Para modelar el punto de contacto de las yemas con los objetos según los parámetros DH se ajustan las yemas de los dedos a esferas de 40mm de diámetro. También se utilizan las referencias ficticias 6, 7 y 8. La articulación ficticia 7 comparte su centro con la numero 6 y su componente en Z es normal al plano formado por z6 y x6 (Figura 4.10). Finalmente se obtiene el marco de referencia 8, con su origen sobre la superficie de la yema del dedo y el vector X8 normal a la superficie de la yema.

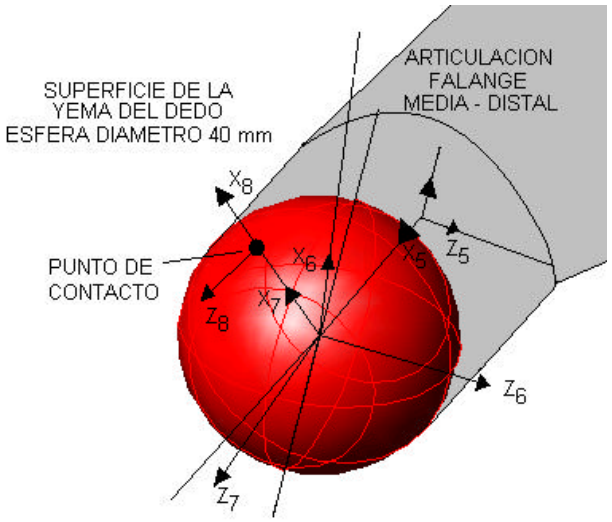


Figura 4.10 – Articulaciones ficticias de las yemas. Fuente: IOC.

El funcionamiento de las articulaciones ficticias es el siguiente:

La primera de las articulaciones (6) permite barrer todos los ángulos de una circunferencia que pertenece al plano de acción del dedo, mientras que la segunda (7) permite barrer todos los ángulos de la circunferencia perpendicular al plano de acción y cuya inclinación es la definida por el ángulo anterior. La tercera referencia ficticia (8) es el resultado de trasladar el centro de la yema hacia la superficie en la dirección definida por las dos articulaciones anteriores. En conclusión, las tres articulaciones ficticias permiten considerar contactos en cualquier punto de la yema del dedo (ver Figura 4.10).

Los parámetros DH de MA-I utilizados en el desarrollo de MAI están calculados según la metodología utilizada por Craig (1989, p. 68). Ver los valores para cada dedo en las siguientes tablas: Tabla 4.1, Tabla 4.2, Tabla 4.3, y Tabla 4.4.

Anular	T01	T12	T23	T34	T45	T56	T67	T78
a [mm]	0	-67	0	76.66	56	33.62	0	20
a [º]	0	90	90	0	0	0	-90	0
d [mm]	197.5	9.5	0	0	0	0	0	0
? [º]	90	78/102	-7.5/82.5	52.5/97.5	-16.7/73.4	20/135	-45/45	0

Tabla 4.1 – Parámetros DH de la cadena Base-Anular.

Medio	T01	T12	T23	T34	T45	T56	T67	T78
a [mm]	0	0	0	76.66	56	33.62	0	20
a [º]	0	90	90	0	0	0	-90	0
d [mm]	197.5	9.5	0	0	0	0	0	0
? [º]	90	78/102	-7.5/82.5	52.5/97.5	-16.7/73.4	20/135	-45/45	0

Tabla 4.2 – Parámetros DH de la cadena Base-Anular.

Índice	T01	T12	T23	T34	T45	T56	T67	T78
a [mm]	0	67	0	76.66	56	33.62	0	20
a [º]	0	90	90	0	0	0	-90	0
d [mm]	197.5	9.5	0	0	0	0	0	0
? [º]	90	78/102	-7.5/82.5	52.5/97.5	-16.7/73.4	20/135	-45/45	0

Tabla 4.3 – Parámetros DH de la cadena Base-Anular.

Pulgar	T01	T12	T23	T34	T45	T56	T67	T78
a [mm]	0	7.56	0	76.66	66	39.17	0	20
a [º]	0	14.11	90	0	0	0	-90	0
d [mm]	264	-203.32	0	0	0	0	0	0
? [º]	-44.56	33.4/57.4	-7.5/82.5	32.5/97.5	-12.7/77.3	20/135	-45/45	0

Tabla 4.4 – Parámetros DH de la cadena Base-Anular.

A partir de los parámetros DH de las cadenas cinemáticas se construyen las matrices de transformación (según Ec. 4.1) que al ser multiplicadas por los puntos del sistema i devuelve un vector con sus componentes en la base i-1. Por lo tanto, si se forman las matrices de transformación con el valor angular ? preciso se pueden obtener la posición y la orientación de los dedos y sus sistemas de referencia asociados.

$$ {}^{i-1}_{i}T = \begin{pmatrix} \cos(q_i) & \sin(q_i) & 0 & a_{i-1} \\ \sin(q_i)\cdot\cos(a_{i-1}) & \cos(q_i)\cdot\cos(a_{i-1}) & \sin(a_{i-1}) & \sin(a_{i-1})\cdot d_i \\ \sin(q_i)\cdot\sin(a_{i-1}) & \cos(q_i)\cdot\sin(a_{i-1}) & \cos(a_{i-1}) & \cos(a_{i-1})\cdot d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{(Ec. 4.1)} $$

Finalmente, cabe destacar que la librería MAI desarrollada es capaz de proporcionar a partir de los valores de las 16 articulaciones de que consta la mano (consultar el rango de valores posibles en la Tabla 4.5), la posición del centro de las yemas de los cuatro dedos.

Articulaciones	0	1	2	3
Dedos de Tipo I (A,M,I)	-12/12	0/90	45/90	0/90
Dedos de tipo II (P)	-12/12	0/90	25/90	0/90

Tabla 4.5 – Rango de movimiento de los dedos de MA-I.

4.3.4. Comunicación con el sistema MR

El control de los movimientos la Mano y del Robot se realiza por separado. Como ya se ha explicado anteriormente, la MA-I se controla mediante el software de control MAIQNX6 y el armario CS8C realiza las tareas de control sobre el TX-90.

No forma parte del alcance de este proyecto unificar las tareas de control, por lo que para poder comandar los movimientos del sistema MR y realizar tareas de aprehensión y manipulación, ha sido necesario desarrollar un sistema de comunicación con la mano y el robot.

Dado que todos los equipos del laboratorio de robótica del IOC y el controlador CS8C son potencialmente capaces de comunicarse mediante sockets, se decidió que este sería el método utilizado para el comando y la comunicación con el sistema MR.

A continuación se introduce el concepto y el funcionamiento básico de la comunicación a través de sockets. Posteriormente se describe el software de comunicación del robot (VMR VAL3) y de la mano (MAIQNX6 Server), y finalmente, se detalla el software de comunicación del VMR.

Socket

Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse datos.

Las propiedades de un socket dependen de las características del protocolo en el que se implementa. Los protocolos más utilizados son el TCP (Siglas del Protocolo de Control de la Transmisión) y el UDP (Siglas del Protocolo de Datagramas de Usuario).

El protocolo utilizado para la comunicación con la mano y el robot es el TCP, dado que tiene las siguientes propiedades [COMER, D. E., 2000, p. 194]:

- Garantiza la transmisión de todos los datos sin errores ni omisiones. Esta característica impide que el sistema MR reciba una contraseña errónea debido al sistema de comunicación.

- Se garantiza que los datos llegan a su destino en el mismo orden en que se han transmitido. Por tanto, al realizar envíos de múltiples datos, se tiene la certeza de que la mano y el robot reciben las consignas según el orden establecido por el software VMR.

- Permite distinguir diversas aplicaciones en una máquina a través del concepto de puerto. En consecuencia, la comunicación entre el programa VMR, la mano y el robot se puede realizar simultáneamente.

El establecimiento de la comunicación mediante sockets se realiza según el modelo cliente-servidor (Figura 4.11). Esta metodología distingue entre dos tipos de sockets, ya que para establecer la comunicación el servidor y el cliente actúan de formas diferentes. El servidor abre el canal de comunicación y permanece a la espera de que el cliente se comunique con él, por lo tanto, el inicio de la comunicación del servidor es pasiva y la del cliente es activa.

Figura 4.11 – Modelo Cliente Servidor.

Los pasos a seguir para utilizar un socket servidor son los siguientes:

1. Crear el socket.
2. Asignar al socket la dirección IP (Número que en Internet identifica a un ordenador que utilice el protocolo "Internet Protocol") y el puerto de conexión que va a utilizar.
3. Escuchar en el puerto del punto anterior y establecer una cola de conexiones entrantes para atender a los clientes según el orden de llegada, o rechazarlos en caso de superar la cantidad deseada por el usuario.
4. Aceptar las conexiones entrantes por el puerto establecido en el paso 2.
5. Enviar y recibir los datos.
6. Cerrar el socket, para liberar el puerto de conexión utilizado.

Para el socket cliente se realizan las siguientes acciones:

1. Creación del socket.
2. Conectar al socket servidor según la dirección IP, y el puerto de conexión que tiene asignado.
3. Enviar y recibir los datos.
4. Cerrar el socket, para liberar el puerto de conexión utilizado.

En la Figura 4.12 se muestra un esquema de la relación temporal entre las acciones que realizan los sockets cliente y servidor.

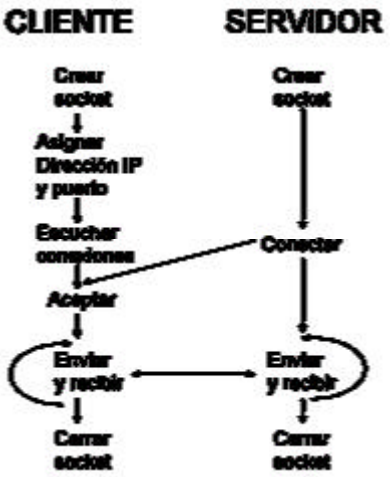


Figura 4.12 – Comunicación mediante sockets.

MAIQNX6 Server

Debido a los proyectos de investigación relacionados con la mano que se realizan en el IOC, el software de la MA-I se encuentra en continuo desarrollo.

En el momento de la realización de este proyecto se estaba realizando la implementación (en el PC de la mano) de un socket servidor que recibiese las consignas definidas por un guante equipado con sensores de posición (instalado en COBE). Finalmente, debido al objetivo común de los dos proyectos se implementó el software del socket servidor conjuntamente con Carlos Rosales y Patrick Grosch. El código resultante se puede consultar en el Anexo D.

Este servidor se integra en el software MAIQNX6 que está escrito en lenguaje C (lenguaje creado en 1969 utilizado en la implementación de numerosos sistemas operativos). La interfase gráfica de usuario ha sido diseñada con la librería Photon, que permite el empleo de ventanas en sistemas operativos QNX.

Esta interfase está formada por cuatro zonas: la zona de la máquina local, la parte informativa de la máquina remota, la zona de control de conexión y la ventana de información (ver Figura 4.13).

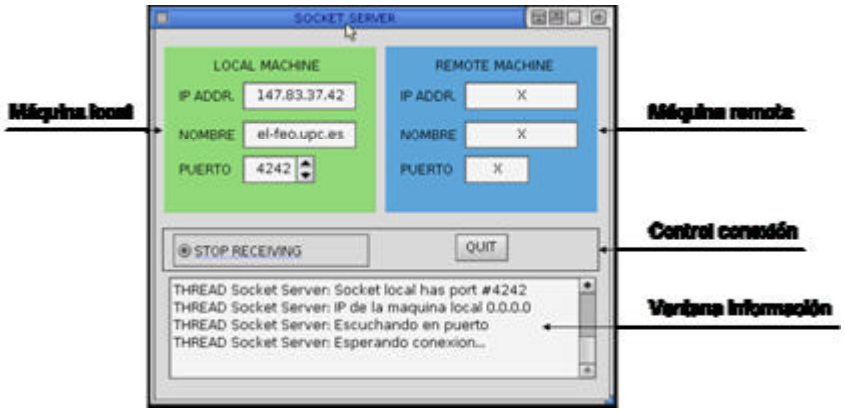


Figura 4.13 – Interfase gráfica del MAIQN6 Server.

En la zona de máquina local se muestra la dirección IP del ordenador en el que se ejecuta MAIQNX6 Server, así como su nombre (en este caso EL-FEO), también se incluye en esta zona un cuadro en la que se puede introducir y cambiar el número de puerto que quiere ser utilizado para el socket servidor.

El cuadro azul corresponde a la zona de la máquina remota, siendo ésta meramente informativa, sirve para advertir al usuario de la dirección IP, el nombre y el puerto del PC que se ha conectado mediante un socket cliente.

La zona de control de conexión está formada por dos botones. El primero (que tiene escrito “STOP RECEIVING” en la Figura 4.13), sirve para indicar al programa si debe iniciar el socket servidor (al iniciar la interfase muestra el rótulo “START RECEIVING”) o cerrarlo. El segundo botón sirve para cerrar la aplicación.

Finalmente, la interfase cuenta con una ventana de información que muestra los procesos que va realizando el programa, y los mensajes que envía y recibe.

El esquema de conexión del MAIQNX6 Server es el que se ha descrito con anterioridad para los sockets de tipo servidor, por lo que no se detallará de nuevo el método utilizado para el establecimiento de la comunicación. Sin embargo, si que resulta necesario describir el tipo de mensajes que debe recibir, cómo los procesa para que la mano ejecute la consigna, y que tipo de mensaje devuelve al cliente.

Tal y como se explica en el subapartado 3.1.3 (Software de la mano), el módulo Pathplanner del MAIQNX6 tiene como valores de entrada la posición inicial y final de las articulaciones, la velocidad de crucero y una aceleración máxima permitida. Con estos parámetros se generan los perfiles de movimiento que son enviados al gestor de consignas y posteriormente son ejecutados por el controlador.

También cabe la posibilidad de que se desee hacer un HOME desde el software VMR, o incluso que se desarrolle un control de fuerza para la mano y se desee comandarlo remotamente.

Para dotar de esta flexibilidad al MAIQNX6 Server se estableció un protocolo de composición de los mensajes que recibe (ver Figura 4.14). Esta convención propone que los mensajes recibidos serán cadenas de un máximo de 128 caracteres, pudiendo ser ampliados en caso de necesidad, aunque se consideran suficientes debido a la resolución angular del control de la mano.

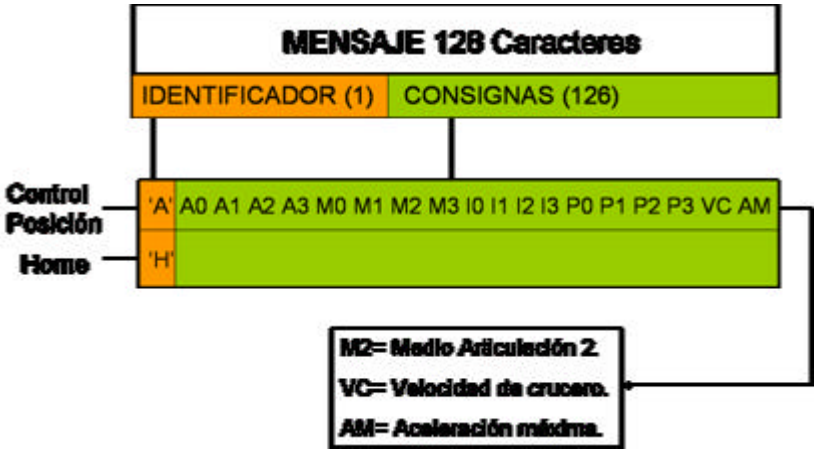


Figura 4.14 – Protocolo de composición de mensajes de MAIQNX6 Server.

El primer carácter del mensaje será un identificador del tipo de control a realizar. Concretamente, para realizar el control de posición se utiliza una 'A' y para realizar el HOME se utiliza una 'H' ambas en mayúsculas. A continuación, en función del tipo de mensaje se añaden los valores necesarios para el Pathplanner. Esto es, en el caso del control de posición, 18 valores enteros. Los primeros 16 corresponden a los valores de las articulaciones de la mano expresados en grados (ver orden en la Figura 4.14), el valor de la velocidad de crucero y finalmente el de la aceleración máxima. En el caso del HOME no es necesario añadir ningún valor, porque éste se realiza utilizando los sensores de final de carrera.

Nótese que en el caso de querer ampliar el control remoto con el envío de consignas de fuerza, sólo sería necesario utilizar un carácter de distinción (por ejemplo 'F') en los mensajes e introducir a continuación la información requerida para el control de fuerza.

Una vez se ha recibido el mensaje, MAIQNX6 Server lo identifica a través del carácter de tipo. Si el mensaje es una consigna de posición articular, el programa extrae las

consignas del mensaje y las transforma en una tabla de 18 enteros. Si el mensaje es de tipo HOME, no hace falta extraer ningún valor de la zona de consignas.

Una vez procesado el contenido del mensaje MAIQNX6 Server lo introduce en la memoria compartida del MAIQNX6 para que éste tome las consignas y realice el control.

Si hay algún error en este proceso o en la recepción del mensaje, el programa envía un mensaje de respuesta formado por una cadena de dos caracteres que contiene la palabra 'KO' en mayúsculas. En el caso de haber funcionado todo según lo previsto, el MAIQN6 Server envía un mensaje del mismo tipo, pero con la palabra 'OK'. De tal modo que cuando el cliente recibe este mensaje, significa que la mano ha recibido la consigna y está realizando el control de la misma.

VMR VAL3

Tal y como se describe en el subapartado 3.2.2 (Armario de Control (CS8C)). El brazo robot del IOC está equipado con un enlace ethernet que puede ser configurado para comunicarse mediante sockets.

La implementación de los sockets que utiliza el CS8C es específica de STÄUBLI, y por lo tanto restringe el tipo de mensajes que son capaces de recibir y enviar. Las limitaciones más importantes son el tamaño de la cadena de caracteres que manejan los sockets (un máximo de 128 caracteres sin posibilidad de ampliación) y el carácter de fin de cadena. Este carácter debe ser seleccionado cuando se crea el socket. Su función es indicar al sistema que la recepción o el envío de un mensaje ha terminado antes de los 128 caracteres.

El desarrollo del software de comunicación con el robot VMR VAL3 se ha realizado de una forma similar a la del MAIQNX6 Server. En este caso, existe un software desarrollado por Emmanuel Nuno llamado Tirol, que permite ejecutar dos servidores simultáneamente. El servidor ServerTele que recibe consignas y el ServerMoni que envía la posición de los ejes del robot según el período deseado.

Para el desarrollo del proyecto, se ha optado por la adaptación del programa Tirol a las necesidades de comunicación del sistema VMR. La mayor diferencia del software adaptado respecto al Tirol es que sólo esta habilitado el servidor de teleoperación, y que éste devuelve un mensaje cuando el robot ha llegado a la posición de consigna. Pese a ello se conserva el código del ServerMoni, en previsión de futuras ampliaciones de funcionalidad. El nuevo programa se conoce como VMR VAL3 y el código relacionado con éste proyecto se encuentra disponible en el Anexo E.

El VMR VAL3 también contiene un protocolo para la composición de los mensajes que recibe. Coincide con el utilizado por Tirol, y consiste en seis valores que corresponden a los ángulos en grados de las seis articulaciones del robot (empezando por la 1 hasta la 6 de la Figura 3.9) seguidos por el carácter '@' que sirve de carácter de fin de cadena. Nótese que a diferencia del Tirol, el VMR VAL3 resta 22 grados al valor recibido de la sexta articulación. Esta resta se realiza con la finalidad de corregir el ángulo de desfase que tiene el adaptador de MA-I respecto la referencia de la muñeca del robot.

Otro de los cambios de VMR VAL3 respecto al Tirol es que una vez el robot se ha situado en la posición deseada, el socket servidor devuelve un mensaje al cliente conectado (en este caso el software VMR), compuesto por la siguiente cadena de dos caracteres en mayúsculas 'OK'.

Clientes VMR

Finalmente, la función de comunicación con el sistema MR se completa con la creación de dos sockets clientes integrados en el software VMR (consultar el código del VMR en el Anexo F) que permiten el envío de consignas y el acuse de recibo de los movimientos.

Dado que uno de los objetivos del proyecto es que el software del sistema VMR sea multiplataforma se han utilizado dos bibliotecas que implementan sockets. La librería winsock de Microsoft para ser utilizada en sistemas operativos Windows, y un paquete de librerías para los sistemas basados en Linux (Ver el Archivo vmr.h en el Anexo F), ambas disponibles en lenguaje C++.

Estas bibliotecas se encuentran integradas en el compilador utilizado en este proyecto GCC, y se utilizan adecuadamente cuando se enlaza el software VMR. Por lo tanto no es necesario realizar ninguna instalación adicional para el uso de sockets en la función de comunicación del software VMR con el sistema MR.

El software diseñado contiene dos sockets integrados disponibles para realizar el envío de consignas a la mano, mediante el socket llamado socketMano, y al robot, a través del socketRobot.

Como ya se ha explicado con anterioridad, los sockets de tipo cliente deben conectarse a la dirección donde se encuentra el servidor, y al puerto que éste tiene asignado para la comunicación. Estos parámetros pueden cambiar con facilidad, por lo que se ha dotado a la interfase del software VMR de herramientas para introducir y modificar estos datos con comodidad (este tema se trata con más profundidad en el subapartado siguiente).

Finalmente, para establecer una comunicación con el servidor deseado es necesario realizar la conexión, y seguir el protocolo descrito anteriormente para la composición de los mensajes enviados y la decodificación de los datos recibidos.

4.3.5. Interacción con el usuario

La función de interacción de usuario se realiza mediante la interfase gráfica del software VMR. Esta interfase permite al usuario introducir datos, ejecutar tareas, así como obtener información sobre el estado del sistema, o visualizar las imágenes capturadas.

El software VMR debe albergar las herramientas de visión, de determinación de consignas, y de comunicación descritas con anterioridad. Dado que todas estas herramientas están programadas en C++, resulta natural que la implementación del software de interacción con el usuario también se haga en este lenguaje. Por otra parte, el objetivo de crear un software multiplataforma afecta directamente a la implantación de la interfase VMR. Para compaginar ambos criterios se ha utilizado la librería Qt de la firma Trolltech.

La librería Qt es una biblioteca multiplataforma diseñada con el objetivo de proporcionar una herramienta para el desarrollo de interfaces gráficas de usuario. La librería está programada en C++, pudiendo utilizarse en entornos Windows y Linux. Qt se distribuye con una licencia GPL para el desarrollo de software de código abierto y software libre. La instalación de la biblioteca Qt depende del sistema operativo en el que se realiza, por lo que se efectuará siguiendo las instrucciones de Trolltech en cada caso.

Las librerías Qt utilizadas en el proyecto se obtuvieron de las siguientes páginas web: http://trolltech.com/developer/downloads/qt/windows (para sistema operativo Windows) el 28 de marzo de 2007 y http://trolltech.com/developer/downloads/qt/x11 (para sistemas como Linux) el 12 de junio de 2007. En estas páginas también se pueden encontrar las instrucciones de instalación de la librería.

El paquete Qt contiene diversas herramientas para facilitar el diseño de las interfases gráficas de usuario. Una de ellas es Qt Assistant, que proporciona una amplia documentación sobre las 400 clases que contiene la librería. En este grupo de herramientas Qt también se incluye la aplicación Qt Designer, que permite diseñar la disposición de los componentes de la interfase de forma gráfica, lo que ayuda a un desarrollo rápido y a una flexibilidad elevada a los cambios.

El uso de Qt Designer es opcional por parte del diseñador, ya que la estructura gráfica de la interfaz también se puede programar directamente con las clases de Qt. Sin embargo, en el diseño del entorno gráfico del VMR se ha utilizado esta aplicación, obteniendo un

fichero llamado *vmr.ui* (disponible en el Anexo F) que contiene la disposición de la interfaz gráfica. Este fichero puede ser modificado a través del Qt Designer intuitivamente y con facilidad por otros programadores, facilitando, por tanto, una futura adaptación de la interfaz por parte de los investigadores que utilizarán la plataforma VMR en la experimentación de algoritmos y estrategias de localización, aprensión de objetos y manipulación.

Nótese que durante la compilación se genera automáticamente el fichero *ui_vmr.h* (disponible en el Anexo F) a partir del *vmr.ui*. Este fichero de tipo cabecera corresponde con el fichero que se editaría manualmente en caso de no utilizar el Qt Designer.

En la Figura 4.15 se muestra la interfase gráfica del VMR. Y a continuación se describen las funciones de las diversas zonas que la componen: la parte de Conectividad (1), el Home o inicialización (2), la zona de Imagen (3), la región de las Tareas (4) y finalmente la zona de Información (5).

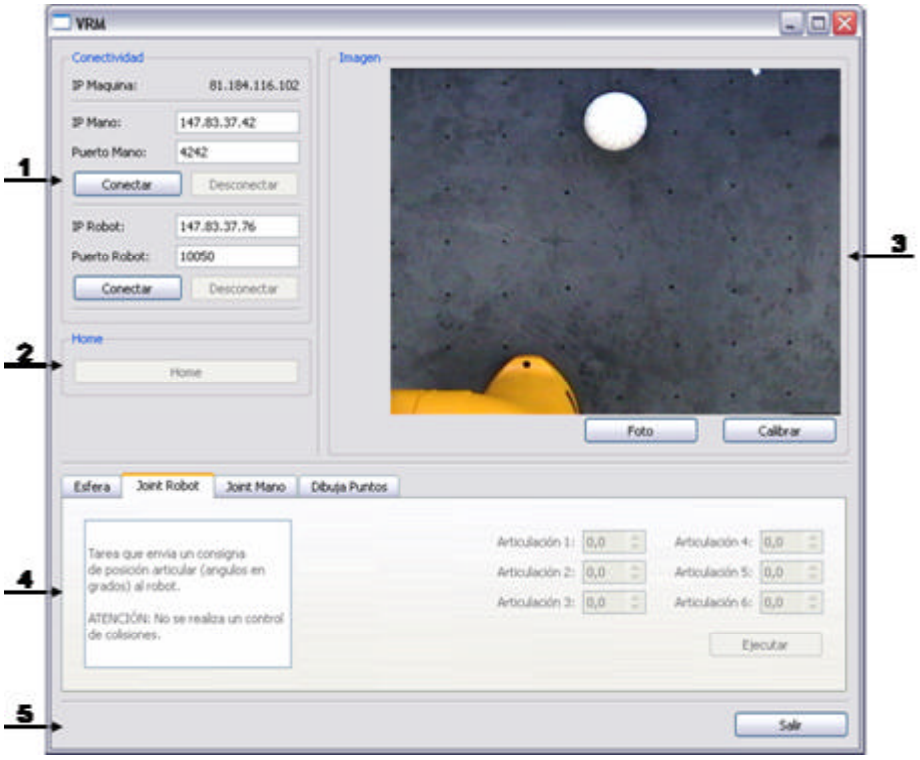


Figura 4.15 – Interfase gráfica del VMR.

Conectividad

La parte de Conectividad (1) de la interfase permite conocer la dirección IP del equipo en el que se está ejecutando el VMR, introducir la dirección IP y el puerto de conexión de los

sockets servidores de la mano y el robot, y finalmente, permite conectar y desconectar con dichos dispositivos.

Los valores que aparecen por defecto en los campos IP y puerto (de la mano y del robot) se inicializan en el fichero *vmr.ui* y pueden ser modificados con facilidad a través del Qt Designer. Esta modificación sólo es interesante en caso de que la dirección o el puerto por defecto de estos servidores cambie de forma permanente, dado que es necesario recompilar el software para que se apliquen los cambios efectuados en el fichero *vmr.ui*. En caso de un cambio momentáneo es preferible modificar los valores desde la interfaz.

Como ya se ha dicho con anterioridad, en esta zona de la interfaz (1) también se presenta la dirección IP del PC local. Esta información puede ser válida en caso de que se desee enviar datos al PC en el que se ejecuta el VMR (por ejemplo, imágenes). El código para obtener estos datos se encuentra en el constructor de la clase vmr (archivo *vmr.cpp* del anexo F). Por lo tanto, esta información se recopila en el momento de abrir el ejecutable *vmr.exe*, siendo válida en caso de trasladar el programa a otro ordenador.

Finalmente, se pueden encontrar cuatro botones de acción en la zona de conexión del VMR (1). Dos botones para conectar, y otros dos para desconectar los sockets de tipo cliente de que dispone el software. El estado por defecto de los botones es activo para los botones de conexión e inactivo para los de desconexión. Esta propiedad se inicializa en el archivo *vmr.ui* aunque cambia cuando se realizan las siguientes acciones (consultar archivos *vmr.cpp* y *vmr.h* del Anexo F):

- Se clica en el botón conectar, y la conexión se realiza con éxito. En este caso el botón conectar pasa a ser deshabilitado, y el botón desconectar activado.

- Falla la conexión durante el envío o recepción de datos. En consecuencia, la conexión que ha fallado (bien la de la mano, o la del robot), vuelve al estado inicial, con el botón conectar habilitado y el desconectar desactivado. De esta forma se puede reintentar la conexión para seguir trabajando.

- Cuando se ejecuta alguna función del VMR. En este caso, toda la interfase permanece inhabilitada para que cualquier demanda del usuario no pueda interferir en la ejecución de la tarea.

Home

La zona de Home o inicialización (2) está formada únicamente con un botón de acción. El estado por defecto de este botón es inactivo (definido en el archivo *vmr.ui*), y su activación, depende directamente de la zona de conexión (1).

Para que el botón Home se habilite, es necesario que se haya establecido la conexión con el socket servidor de la mano, o bien con el del robot. En este caso se podrá realizar la función Home que lleva al robot a una posición conocida (el valor de todas sus articulaciones a 0º) y que realiza remotamente el HOME de la mano.

Del mismo modo, el botón Home puede quedar deshabilitado si se pierde la conexión con la mano y el robot, o se desconectan deliberadamente (ver archivo *vmr.cpp* y *vmr.h* en el Anexo F).

Imagen

La zona de Imagen de la interfase VMR (3) contiene dos botones (Foto y Calibrar) y un recuadro donde se muestran las imágenes adquiridas (en algunos casos también procesadas) del sistema de visión. Los dos botones se encuentran siempre en estado activo ya que no requieren ningún tipo de acción previa para su utilización, ni existe ningún estado del software VMR que requiera que no se utilicen estas funciones.

El botón Foto realiza una foto y la muestra en el recuadro de las imágenes adquiridas. Para realizar esta acción se utiliza el programa Captura, pero no es necesaria la biblioteca Computer Vision (ver archivo *vmr.cpp* y *vmr.h* del Anexo F).

La región de Imagen no se muestra completa en la Figura 4.15, ya que existe un panel que aparece y desaparece al hacer clic sobre el botón calibrar. La apariencia de esta región al completo se muestra en la Figura 4.16. El código necesario para ocultar y mostrar el panel así como el resto de funciones de la zona Imagen se encuentra disponible en los archivos *vmr.cpp, vmr.h* y *vmr.ui* del Anexo F.

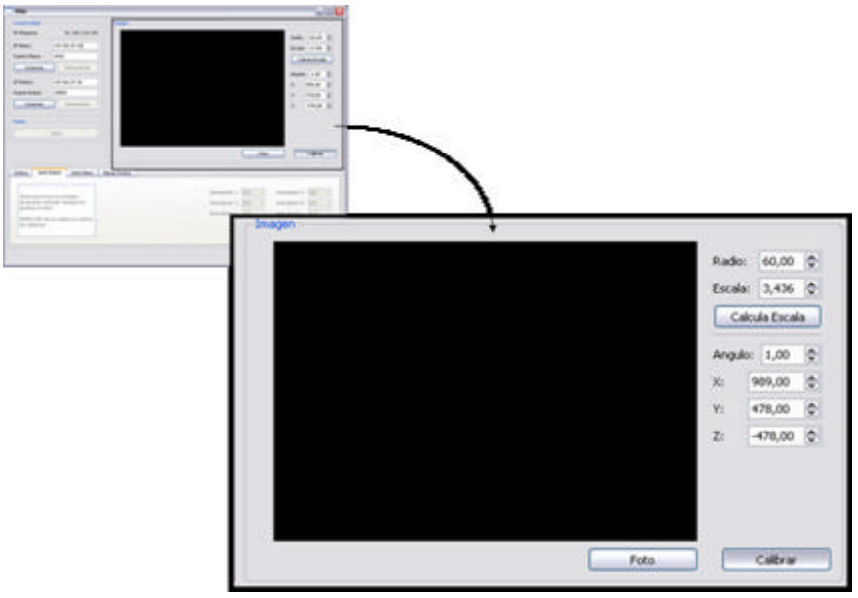


Figura 4.16 – Zona de imagen al completo en la interfase VMR.

El motivo de que el panel se encuentre oculto por defecto es que la cámara ya se encuentra calibrada y que un cambio por error en los valores puede hacer que el robot colisione con la mesa. Por lo tanto, el panel permite la calibración en caso de que la situación de la cámara (o la mesa) respecto al robot se modifique, manteniendo el compromiso con la seguridad durante el funcionamiento normal del sistema.

En el panel desplegado por el botón calibrar se pueden distinguir dos zonas. La primera contiene dos selectores numéricos y un botón llamado Calcula Escala, y la segunda está formada por cuatro selectores del mismo tipo.

La primera zona es una utilidad que sirve para calcular la escala con la que se observa un objeto en la mesa de trabajo. Para ello, se debe introducir el radio en milímetros de la circunferencia que se colocará en la mesa, y presionar el botón calcular escala. Se toma una imagen, de la que se obtiene el radio en píxeles de la circunferencia. Con estos datos y suponiendo que la orientación de la colocación de la cámara es la descrita en el subapartado 4.2.4, se calcula el valor de escala y se actualiza en el selector numérico correspondiente. También es posible introducir una escala manualmente.

La segunda parte de la zona permite situar la imagen y corregir la rotación de la cámara respecto de los ejes del robot (ver Figura 4.17) que se toman como referencia absoluta. El primer valor a introducir es el ángulo de giro de los ejes de la imagen (amarillo) respecto de los ejes del robot (azul) expresado en grados (a). Los tres siguientes valores son la posición del origen de la imagen (esquina superior izquierda, valores x, y, z, de la Figura 4.17) respecto la referencia absoluta (consultar subapartado 4.3.3 donde se describe el sistema de referencia del robot).

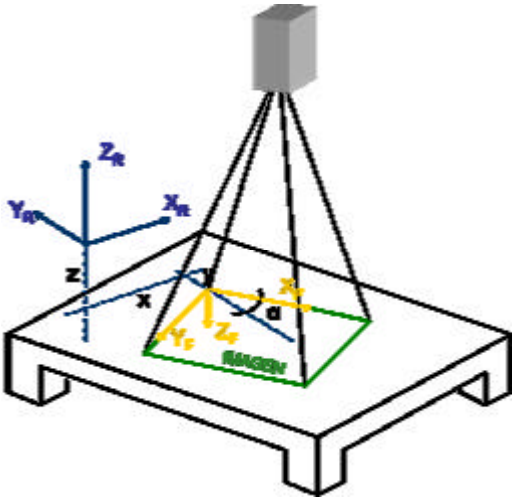


Figura 4.17 – Parámetros para la calibración de la imagen.

Los valores de todos los cuadros numéricos se inicializan en el archivo *vmr.ui*, y se pueden establecer a través del Qt Designer diversas propiedades de los cuadros como el rango, el incremento de las flechas y la precisión de los mismos.

Tareas

La zona número 4 de la Figura 4.15 se reserva para las tareas que utilizan el software VMR. En la Figura 4.18 se puede observar que la zona está compuesta por cuatro tareas diferentes: Esfera (1), Joint Robot (2), Joint Mano (3), y Dibuja Puntos (4).

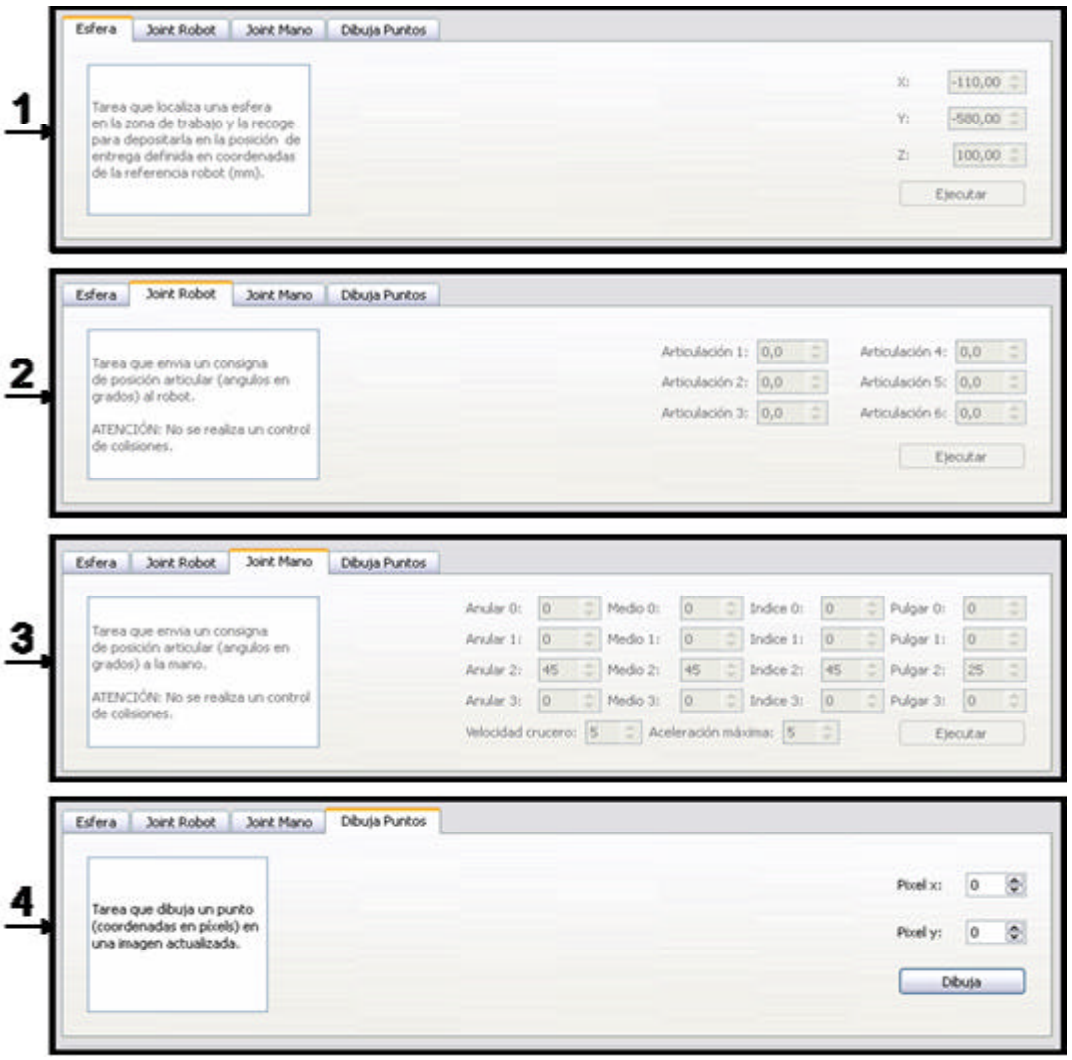


Figura 4.18 – Zona de tareas de la interfase VMR.

Para seleccionar la tarea que se desea ejecutar, basta con hacer clic en la pestaña adecuada. Al hacerlo, se muestra una descripción de la tarea, y los elementos necesarios para su funcionamiento. Nótese que el orden de aparición de las pestañas se puede intercambiar con facilidad desde el Qt Designer, sin necesidad de alterar el código manualmente.

El funcionamiento de las tareas se detalla en el Capítulo 5, pero para demostrar la capacidad de la interfase gráfica en el control de las acciones del usuario, se describen las condiciones necesarias para la habilitación de cada una de las aplicaciones (código disponible en los archivos *vmr.cpp*, *vmr.h,* y *vmr.ui* del Anexo F). En el caso de que alguna de las condiciones no se cumpliera, la interfaz deshabilita las tareas afectadas, para mayor seguridad del sistema.

El estado inicial de las tareas es deshabilitado para las aplicaciones Esfera, Joint Robot y Joint Mano. Sin embargo, la aplicación Dibuja Puntos está siempre disponible. Esto es debido a que Dibuja puntos no requiere ningún tipo de inicialización ni paso previo.

En el caso de Joint Robot es necesario realizar el Home del robot con éxito y en consecuencia haber conectado con el socket servidor del robot VMR VAL3 previamente. Análogamente, la función Joint Mano requiere la conexión de la mano con MAIQNX6 Server y la realización del Home con éxito para inicializar el servicio. Finalmente, para la tarea Esfera, se requiere que ambas conexiones estén activas y que el Home del sistema MR se haya efectuado con anterioridad.

Finalmente, solo cabe destacar que el código de la interfase VMR, no reconoce las tareas por el nombre que se muestra en pantalla, sino por la palabra Tarea más el número correspondiente al desarrollo de la misma. Por ejemplo, la Tarea2 se corresponde con Joint Mano, ya que esta fue la segunda tarea en ser implementada. Esto se debe a que el nombre a mostrar en la interfase suele cambiar con facilidad.

Información

La zona de información corresponde al número 5 de la Figura 4.15. Esta zona está disponible para insertar mensajes informativos en todo momento, y también cuenta con una barra de progreso para informar al usuario en el transcurso de una operación.

Inicialmente esta zona se encuentra oculta al usuario según se ha establecido en el archivo *vmr.ui* mediante el Qt Designer. En la Figura 4.19 se muestra la utilidad de la zona de Información durante la conexión con la mano. Mientras se muestra un mensaje de espera para el usuario, se informa del estado de la aplicación con la barra de progreso. El código se encuentra disponible en el archivo *vmv.cpp*, *vmr.h* del Anexo F.

Figura 4.19 – Zona de información de la interfase VMR.

4.3.6. Procedimiento para la implantación de una tarea en VMR

El software del sistema VMR está diseñado para servir de plataforma en las investigaciones experimentales que se realizan en el IOC. Para agilizar el aprendizaje en el uso del VMR por parte de los investigadores se aporta al proyecto el siguiente procedimiento.

El proceso de implementación de una tarea en el software VMR consta de las siguientes etapas:

- Creación de la interfase gráfica de la tarea y adaptación a la interfase VMR.
- Desarrollo del código de la tarea.
- Compilación del software VMR.

Para la creación y la adaptación de la interfase al software VMR desarrollado, resulta de utilidad trabajar con la aplicación Qt Designer ya que esta permite un alto grado de flexibilidad y de realimentación visual con el trabajo que se está realizando.

En primer lugar será necesario abrir el archivo *vmr.ui* disponible en el Anexo F mediante el Qt Designer. A continuación se introduce una pestaña nueva en el grupo de tareas haciendo clic derecho sobre una pestaña y seleccionando “Insert Page / Alter current Page” tal y como se muestra en la Figura 4.20.

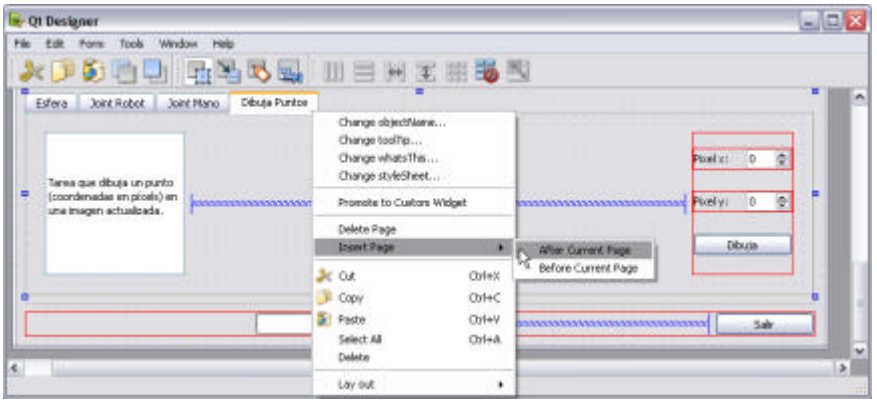


Figura 4.20 – Introducción de una nueva pestaña en el cuadro de tareas del software VMR.

El interior de esta pestaña debe contener un solo Widget (en la terminología utilizada por Qt, se refiere a un contenedor de elementos gráficos) que se puede diseñar según las necesidades de la tarea mediante los widgets que proporciona la biblioteca, o incluso con widgets personalizados.

Para mantener el formato establecido será necesario nombrar a este widget, y a los elementos en él contenidos, mediante el nombre que da Qt al tipo de widget empleado, un descriptor opcional (útil en caso de repetición del mismo tipo de widgets en una misma tarea), y la palabra “Tarea+número de tarea” al final del nombre. En la Figura 4.21 y la Figura 4.22 se muestran algunos ejemplos de nombres definidos mediante el cuadro Property Editor del Qt Designer.

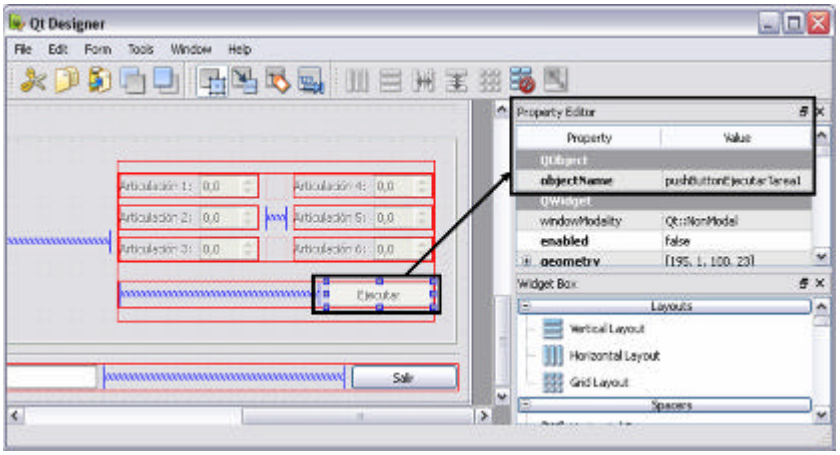


Figura 4.21 – Nomenclatura de los widgets de las tareas implementadas (1).

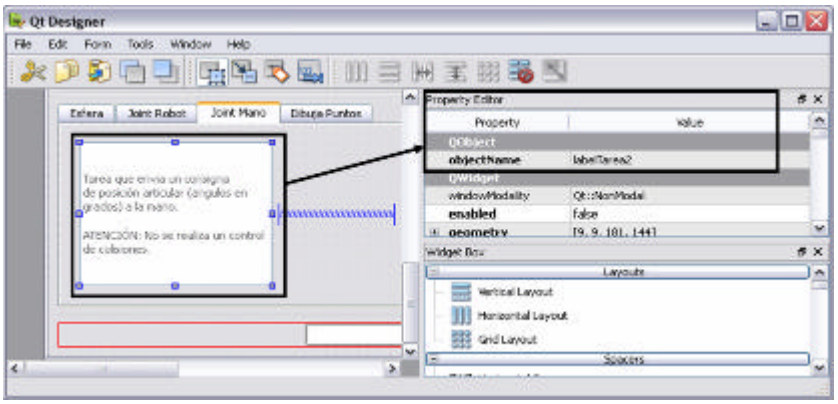


Figura 4.22 – Nomenclatura de los widgets de las tareas implementadas (2).

El desarrollo del código se realiza básicamente en los archivos *vmr.h* y *vmr.cpp* del software VMR. En el archivo *vmr.h* se deben declarar las cabeceras de las funciones que utiliza la nueva tarea, así como las variables globales necesarias.

La declaración de las funciones que se deben ejecutar al interactuar con algún elemento de Qt se debe realizar según su protocolo. Según esta convención el nombre de la función implementada debe seguir el siguiente formato: "on_nombre del widget_acción sobre el widget". Por ejemplo, en el caso de la definición de la función que debe ejecutarse al clicar el widget de la Figura 4.21 "pushButtonEjecutarTarea1". Se debe declarar la función con el siguiente nombre "on_pushButtonEjecutarTarea1_clicked". De esta forma, cuando se realice un clic en este botón se ejecutará el código de esta función.

Existe otro tipo de convención para la programación de las tareas que implican movimiento debido a motivos de seguridad. Esta plantilla se puede consultar en el archivo *vmr.cpp* del Anexo F, aunque la parte fundamental es el uso de las funciones *IniciaTarea()* y *FinalizaTarea()* al inicio del código de las tareas y al fin del mismo respectivamente. Estas funciones impiden que la interfase se mantenga receptiva a las demandas del usuario durante la ejecución de la tarea, impidiendo por lo tanto, posibles interacciones peligrosas.

Para la programación de las tareas pueden necesitarse las funciones de las bibliotecas Qt, MT, TXRobot, MAI, Computer Vision, etc. Para ello, tan solo es necesario utilizar el espacio de nombres de la MT (mt), la TXRobot (robot) y la MAI (mai) como se puede observar en los archivos *vmr.h* y *vmr.cpp*. Nótese que también se ha implementado la clase QvmrSleep que permite realizar una pausa de duración variable con resolución de microsegundos, para la que hay que utilizar el espacio de nombres QvmrSleep (consultar archivo q*vmrsleep.h* del Anexo F).

Finalmente se procede a la compilación del software con el compilador GCC. Este proceso requiere la creación de un directorio llamado *vmr* que contenga los siguientes archivos: *main.cpp, vmr.cpp, vmr.h, vmr.ui, txrobot.cpp, txrobot.h, mai.cpp, mai.h, qvmrsleep.h.* En este directorio se generan archivos automáticos y se localiza el ejecutable una vez compilado el software.

A continuación se ejecutan los siguientes pasos:

1. Ejecutar el comando *qmake -project main.cpp vmr.cpp txrobot.cpp mai.cpp* en un terminal (sistemas linux) o en la línea de comandos (windows). Esta acción genera un archivo *vmr.pro* (al estar en el directorio *vmr*) que será modificado en los pasos 2 y 3, y que se encuentra disponible en el Anexo F.

2. Añadir al archivo *vmr.pro* la libreria de sockets para Windows en el caso de ser éste el sistema operativo. Para hacerlo, introducir al final del archivo el siguiente texto:

   win32 {

   LIBS += -lwsock32

   }

3. Añadir al archivo *vmr.pro* la libreria Computer Vision de Intel introduciendo al final del archivo el texto:

   LIBS += -lhighgui -lcxcore –lcv

4. Ejecutar el comando *qmake vmr.pro*. Esta operación genera los archivos *Makefile*, *Makefile.Debug* y *Makefile.Release*, que utiliza el GCC para optimizar el proceso de compilación.

5. Ejecutar el comando *make* para compilar una versión de depurado del software, o *make release* para obtener una versión de lanzamiento.

5. Aplicación del sistema VMR desarrollado

El diseño del sistema VMR se ha realizado a fin de proporcionar una plataforma de pruebas experimentales de diferentes algoritmos y estrategias de localización, aprehensión y manipulación de objetos. Para verificar su operatividad, se ha aplicado el sistema VMR en una tarea real de localización y manipulación de objetos.

En primer lugar se realiza una descripción de la tarea que debe realizar el sistema VMR, a continuación se detalla el proceso de diseño de la solución, y finalmente se muestran los resultados derivados de la experimentación.

5.1. Descripción de la tarea

La tarea que realiza el sistema VMR se denomina ESFERA y consiste en localizar una esfera de radio variable en el entorno de trabajo, aprehenderla y finalmente depositarla en el lugar determinado por el usuario.

Debido a las limitaciones geométricas y mecánicas de MA-I, el diámetro de las esferas que aprehende ESFERA puede variar entre los 110 y los 160mm.

El entorno de trabajo en el que de desarrolla la tarea ESFERA está formado por la superficie rectangular de 900x400mm situada sobre el plano de la mesa que sirve de soporte para el TX-90. En la Tabla 5.1 se muestran las coordenadas de los puntos que forman la superficie rectangular en la referencia absoluta (que coincide con la referencia del robot descrita en el subapartado 4.3.3).

Puntos	A	B	C	D
X [mm]	839	439	439	839
Y [mm]	328	328	-672	-672
Z [mm]	-478	-478	-478	-478

Tabla 5.1 – Coordenadas de los vértices de la zona de trabajo de ESFERA.

La zona habilitada para la entrega de las esferas, se encuentra delimitada por el cubo definido por los planos expresados según la referencia absoluta que se muestran en el siguiente grupo de ecuaciones.

$$\begin{aligned} x &= -800 \quad x = 800 \\ y &= -800 \quad y = 800 \\ z &= 100 \quad z = 400 \end{aligned} \qquad \text{(Ec. 5.1)}$$

Nótese que esta región incluye una parte del robot, por lo que el usuario debe evitar entregar las esferas dentro de un cilindro vertical de 350mm de radio y concéntrico con la base del robot, a fin de evitar posibles colisiones (ver Figura 5.1).

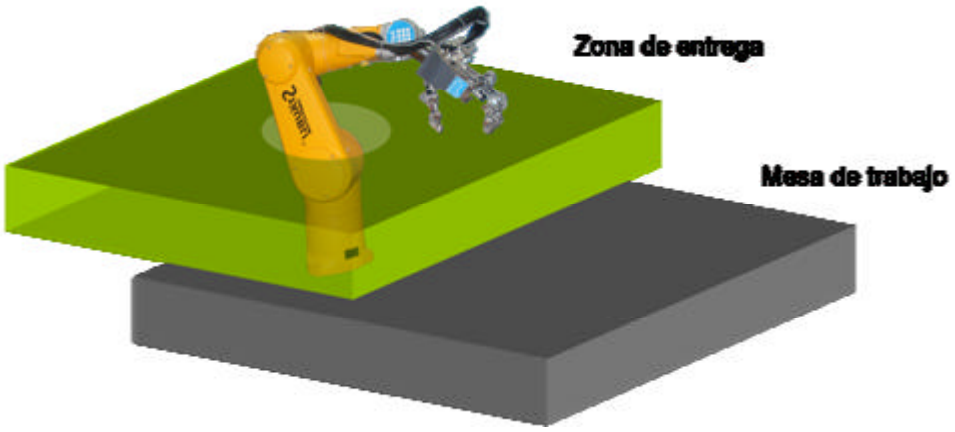


Figura 5.1 – Zona de entrega de ESFERA.

La argumentación para el establecimiento de las especificaciones de ESFERA descritas con anterioridad, se desarrolla con más profundidad en el apartado 5.3 de este documento, ya que algunas derivan de los conocimientos adquiridos durante el diseño y la experimentación real de ESFERA.

5.2. Diseño e implementación

El software que permite realizar la tarea ESFERA se encuentra recogido en los archivos *vmr.cpp* y *vmr.h* del Anexo F, y corresponde a la tarea 3. De todos modos, resulta de interés remarcar que en el citado archivo *vmr.cpp* existen otras tres tareas implementadas (las tareas 1, 2 y 4).

Las tareas 1, 2 y 4 son operativas y se han desarrollado necesariamente durante el diseño de ESFERA. Sin embargo, tienen la entidad suficiente como para formar por si solas una

tarea de utilidad para el usuario. Por lo tanto, quedan a disposición del IOC para ser utilizadas y ampliadas en caso de necesidad.

A continuación se describen brevemente las tres tareas auxiliares del VMR:

- Tarea1. Esta aplicación envía una consigna de posición articular al brazo robot TX-90. Esta tarea se llama Joint Robot en la pestaña de la zona de tareas de la interfase del VMR. Para su correcta ejecución es necesario una conexión activa con el robot y haber realizado el Home con anterioridad. Los valores de las seis articulaciones se introducen en grados y su rango está limitado por el software VMR según las especificaciones de STAÜBLI. Nótese que la nomenclatura de las articulaciones sigue la numeración establecida en la Figura 3.9 (página 19).

- Tarea2. El nombre de esta tarea es Joint Mano en la interfase VMR. Su función es enviar una consigna de posición articular a MA-I. Igual que en la Tarea1, es necesario una conexión activa con la mano y haber realizado el Home con anterioridad. Los datos de las 16 articulaciones se introducen en grados y su rango está limitado por el software VMR para proteger la mecánica de la mano. La interfase de la Tarea3 también permite introducir los valores de la velocidad de crucero y la aceleración máxima (también dentro del rango que les corresponde). La nomenclatura de las articulaciones sigue la numeración establecida en la Figura 3.4 (página 13).

- Tarea4. Esta tarea denominada Dibuja Puntos captura una imagen de la cámara y dibuja un punto rojo en las coordenadas introducidas por el usuario. Los valores de las coordenadas se deben introducir en píxeles. Nótese que la resolución de las imágenes mostradas en la interfase VMR es de 384x288 píxeles.

Como se puede observar, las tareas 1 y 2 sirven para comandar el sistema MR, por lo tanto, desde estas sencillas aplicaciones se pueden dirigir los movimientos de MA-I y el Robot de forma remota y unificada.

Por su parte, la tarea 4, puede ejecutarse en cualquier situación ya que no requiere ningún tipo de conexión. Actualmente puede utilizarse para realizar comprobaciones del calibrado de la cámara, aunque su funcionalidad es fácilmente ampliable mediante las herramientas que contiene la biblioteca Computer Vision.

La interfase de la Tarea3 sigue el protocolo establecido para la zona de tareas. Los únicos valores que el usuario puede introducir en la aplicación son las coordenadas donde se debe entregar la esfera, por lo que la interfase sólo cuenta con tres selectores numéricos y el botón Ejecutar (consultar archivo *vmr.ui* del Anexo F y la Figura 5.2).

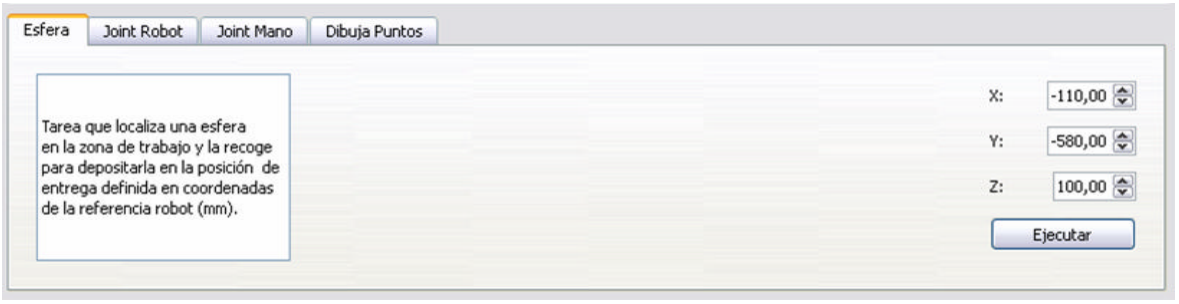


Figura 5.2 – Interfase gráfica de usuario de la tarea ESFERA.

A continuación se describe la estrategia adoptada por la Tarea3 para localizar una esfera, aprehenderla y manipularla. Posteriormente también se explican las principales funciones que ejecuta el software para realizar cada acción.

La metodología que sigue ESFERA para realizar su cometido es similar a la empleada por el ser humano cuando debe realizar una tarea de localización, aprehensión y manipulación de objetos. La ejecución de este tipo de labores consta principalmente de tres fases:

- En primer lugar se localiza el objeto. Distinguiéndolo de entre el entorno mediante la información que recibe el sistema. Nótese que para ello es necesario conocer ciertas cualidades del objeto a localizar.

- En segundo lugar se determina la posición de las extremidades para recogerlo y manipularlo. En este punto es necesario conocer el sistema de recogida para determinar una posición de aprehensión. También es habitual el uso de suposiciones derivadas de la experiencia previa en la aprehensión o deducciones a partir del conocimiento del objeto.

- Finalmente se realizan los movimientos de aprehensión y manipulación determinados en la fase anterior. Normalmente esta fase consta de movimientos de aproximación antes de realizar la recogida del objeto.

Definida la estrategia general a utilizar para realizar la tarea, se procede a describir las acciones que realiza el software de la Tarea3. El código ejecutado al pulsar el botón de la tarea ESFERA corresponde a la función on_pushButtonEjecutarTarea3_clicked(). Este código se encuentra disponible en el archivo *vmr.cpp* del Anexo F. Seguidamente, se explican (en orden de ejecución y en ausencia de errores provocados por el incumplimiento de las especificaciones) las principales acciones que realiza esfera:

1. IniciaTarea. Esta función pertenece al protocolo de tareas que implican movimiento ya que inhabilita la interfase durante la ejecución de la tarea.

2. CapturaValoresCalibrar. Esta función captura los valores actuales del panel de calibrado y los almacena en las variables globales *ImgRefWorld, escalaImg* y *alpha*.

3. A continuación, se inicializa la configuración del robot como lefty, epositive, y wnegative. Esta configuración se ha escogido porque impide la colisión del brazo con la mesa para los puntos que se encuentran por encima de ella.

4. EncuentraCírculo. Esta función captura una imagen de la cámara y localiza una esfera (mediante una función de la librería Computer Vision) devolviendo su posición y radio en píxeles según la referencia Imagen descrita en la Figura 4.17.

5. Seguidamente se procede a la inicialización de la variable sphereRefPhoto que contiene los valores (en milímetros) del centro de la esfera en la referencia del punto anterior. Nótese que se aproxima el radio del círculo encontrado al radio de la esfera, despreciando la perspectiva y la deformación de la cámara.

6. IniciaTransformacionesTarea3. Esta función inicia las transformaciones necesarias (llamadas WorldToSphereEntrega, SphereToHand, HandToWrist, WorldToPhoto, WorldToSphereRecogida, WorldToSphere) para obtener la posición y la orientación de la muñeca del robot en los puntos de recogida, seguridad y entrega. Al final de esta lista se define con más detalle la inicialización y el uso de dichas transformaciones.

7. Cálculo de la cinemática inversa del robot para obtener la consigna articular correspondiente a la posición y orientación de la muñeca en los puntos de recogida, seguridad y entrega.

8. CalculaConsignaManoTarea3. Esta función calcula las posiciones articulares de la mano para coger la esfera, según el diámetro de la esfera obtenido en el punto 4 de la lista. Consultar la metodología utilizada en el apartado siguiente.

9. A continuación se inicializan los mensajes (según el protocolo establecido) que se van a mandar al MAIQNX6 Server y al VMR VAL3, para ejecutar los movimientos determinados en los dos puntos anteriores.

10. EnviaConsigna. Esta función envía una consigna articular determinada a la mano o al robot según corresponda y bloquea el sistema hasta su ejecución. Una vez realizado el movimiento se pide la autorización del usuario para efectuar el

siguiente movimiento o abortar la ejecución de ESFERA. Las consignas enviadas son las siguientes: punto de seguridad, punto de recogida, aprehensión de la esfera, punto de seguridad, punto de entrega, extensión de la mano (con la consecuente liberación de la esfera).

Las transformaciones geométricas inicializadas en el punto 6 se forman a partir de las coordenadas del origen y de los vectores (en este caso se han utilizado bases ortonormales) que forman la referencia i+1 expresados en la referencia i. Las matrices se construyen según se muestra en la (Ec. 5.2). Donde $Z_{i+1}x_i$ corresponde a la componente en X del vector z que forma la base i+1 expresado en la base i. Nótese que las tres primeras filas de la cuarta columna corresponden a las coordenadas del origen de la base i+1 en la referencia i.

$$ {}^{i}T_{i+1} = \begin{pmatrix} X_{i+1}x_i & Y_{i+1}x_i & Z_{i+1}x_i & O_{i+1}x_i \\ X_{i+1}y_i & Y_{i+1}y_i & Z_{i+1}y_i & O_{i+1}y_i \\ X_{i+1}z_i & Y_{i+1}z_i & Z_{i+1}z_i & O_{i+1}z_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \text{(Ec. 5.2)} $$

Las transformaciones se utilizan para obtener la posición y la orientación de la muñeca del robot. Esta información se encuentra disponible en la transformación WorldToWrist (WTW) ya que ésta contiene las coordenadas del origen y la orientación de la base de la muñeca del robot en la referencia absoluta (coincidente con la del robot).

La obtención de la transformación WTW se realiza básicamente mediante la WorldToSphereRecogida (WTSR), la WorldToSphereEntrega (WTSE) y la SphereToWrist (STW). Una vez calculadas, se multiplica WTSR o la WTSE por STW y se obtiene WTW. En este momento, se extrae la rotación y la translación de la transformación, se realiza la cinemática inversa y se obtienen las consignas de recogida y entrega.

La transformación fundamental para la obtención de la STW es la SphereToHand (STH). Esta transformación sitúa la posición y la orientación de la base de la mano a partir de una referencia ficticia situada en la esfera. Para su definición se genera, en primer lugar, la transformación HandToSphere (HTS) y posteriormente se invierte la matriz para obtener la STH.

Para generar HTS se parte de una configuración conocida de la mano, en la que MA-I tiene todos sus dedos extendidos. Utilizando la biblioteca MAI, se calculan las coordenadas del centro de las yemas en la referencia mano. Con estos valores se calcula el origen de la referencia esfera, que se sitúa en el punto medio del segmento definido por la punta de los dedos medio y pulgar, mientras que la base se forma tan solo con los

vectores z, e y, ya que se trata de una base ortonormal. El vector y de la base corresponde al vector normalizado que va de la punta del dedo anular a la del índice. El vector z, es un vector normal al plano formado por las puntas de los dedos anular, pulgar e índice y se orienta hacia la palma de la mano (ver Figura 5.3).

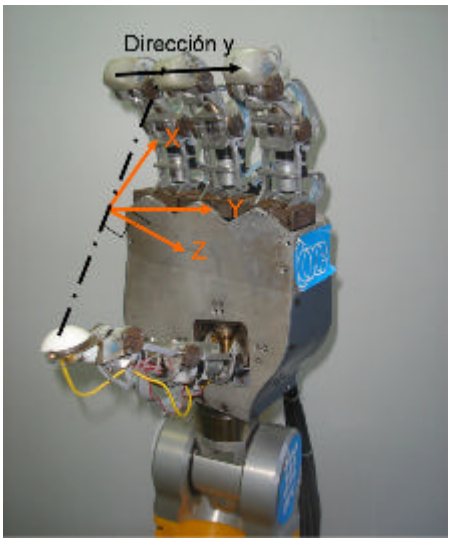


Figura 5.3 – Transformación HandToSphere.

El motivo de construir la transformación con la palma de la mano extendida es que al situar la referencia virtual adecuadamente, la mano no colisionará con la mesa ni la esfera al situarse en la posición de recogida. Para ello sólo resulta necesario situar el origen de la referencia virtual en el centro de la esfera, y posicionar los vectores x, e y, de la base en el plano meridional de ésta según la orientación deseada. En consecuencia las yemas de los dedos se situarán según se muestra en la Figura 5.4.

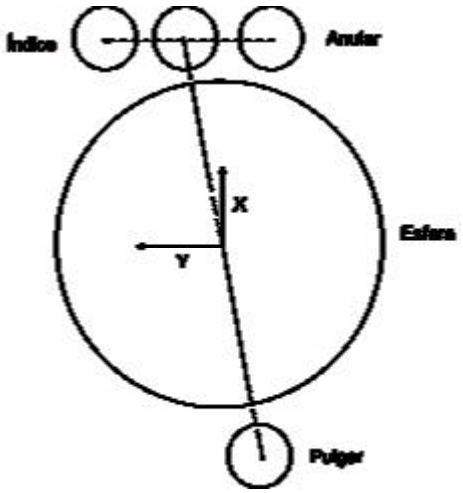


Figura 5.4 – Posición de las yemas respecto la base de la esfera, en el plano Z=0.

Como ya se ha explicado con anterioridad, existe un adaptador que fija la mano a la muñeca del robot. Dado que las bases de la muñeca y la mano tienen la misma orientación, la transformación HandToWrist sólo debe trasladar origen de la referencia 18mm (grosor del adaptador) en la componente z.

Una vez generadas las matrices de transformación anteriores se obtiene SphereToWrist al multiplicar STH por HTW.

En el caso de la transformación WTSR es necesario construir la referencia ficticia de la esfera en la referencia global, para que el robot se sitúe convenientemente al recogerla. Par ello se utiliza la transformación WorldToPhoto (WTP) y las coordenadas de la esfera obtenidas en el punto 4 de la lista anterior.

La transformación WTP se obtiene a partir de los parámetros del panel de calibrado de la cámara. Para ello, se expresan las componentes de los vectores que forman la base de la foto en función de la referencia absoluta (a partir del ángulo a), y se realiza la misma acción con el origen de la referencia (ver esquema en la Figura 4.17 de la página 61). Una vez obtenida esta transformación, se obtienen las coordenadas de la esfera en la referencia absoluta multiplicando WTP por el vector de posición del centro en la referencia imagen.

A partir de los valores del centro de la esfera en la referencia absoluta se puede calcular la orientación de la base (el origen de la base es el propio centro de la esfera) para que la mano la recoja de forma adecuada. Esta información se almacena en la transformación WTSR. Para ello se hace coincidir el vector z de la base con su componente análoga de la referencia global, y se calcula el vector x en la referencia global según el siguiente grupo de ecuaciones, siendo x, e y, los valores de las coordenadas X e Y del centro de la esfera en la referencia global calculados con anterioridad.

$$\vec{x} = (\cos(a+b),\quad \sin(a+b),\quad 0)$$
$$a = \arctan(y, x) \qquad \text{(Ec. 5.3)}$$
$$b = \frac{80 \cdot \Pi}{180} \cdot \frac{900 - \sqrt{x^2 + y^2}}{900 - 400}$$

Como se puede observar en la ecuación (Ec. 5.3), la orientación de esta base ficticia depende de a y ß. El valor de a orienta la mano para que recoja la esfera con el vector que va del dedo anular al índice perpendicular al segmento que se puede trazar del origen de

la referencia global al centro de la esfera, y el ángulo ß modifica esta orientación según la distancia que hay desde el centro de la esfera al eje Z de la referencia absoluta.

Como se puede observar en la Figura 5.5, el giro provocado por ß disminuye a medida que el centro de la esfera se aleja de la referencia absoluta. Siendo ß igual a 0º cuando la distancia es de 900mm y 80º cuando es de 400mm. Esta orientación variable se ha diseñado como una estrategia más de seguridad para evitar colisiones del antebrazo del robot con la base. Nótese que en la Figura 5.5 se muestran las yemas de los dedos de seis posiciones de la mano (coloreadas en verde), que se sitúan según los valores de a y ß para recoger la esfera localizada en seis posiciones distintas.

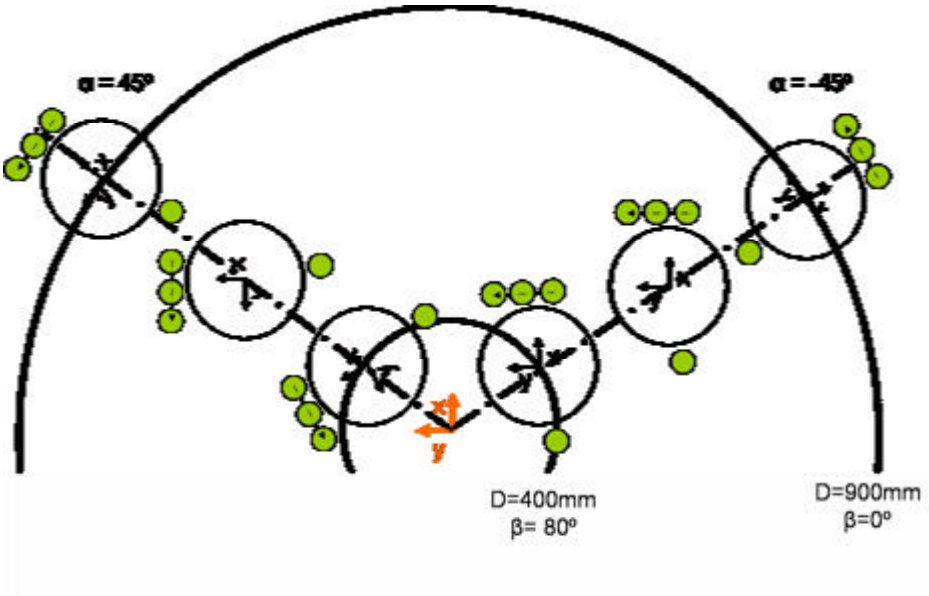


Figura 5.5 – Orientación de la mano en función de la posición de la esfera.

La transformación WTSE se define a partir de los valores introducidos por el usuario. Estos valores corresponden a las coordenadas del centro de la esfera en la referencia global, por lo que se sitúa el origen de la referencia ficticia según las componentes introducidas, y su base se forma a partir del vector x, que tiene la dirección del vector que va del origen de la base al punto de entrega, y el vector z que tiene la misma dirección que su homónimo en la referencia global.

El sistema se encuentra en disposición de obtener la transformación WTW en el caso de la recogida y de la entrega, para ello basta con multiplicar WTSR por STW y WTSE por STW respectivamente.

Por último es necesario calcular la transformación WTW de seguridad, que es igual a la WTW de recogida pero restando 200mm a la componente z del origen de la referencia de

la muñeca. Esta transformación se utiliza a fin de evitar posibles colisiones con la esfera u otros objetos.

5.3. Experimentación

En este apartado se muestran los resultados derivados de la experimentación y el procedimiento a seguir para el uso de ESFERA.

Debido a que la tarea ESFERA debe aprehender esferas con diversos radios, se ha modelado la consigna de las articulaciones en función del diámetro. Para ello, se ha situado la mano en la posición de recogida correspondiente a tres esferas (de diámetro 110, 150 y160mm) y se han establecido consignas manuales (mediante la tarea Joint Mano) para aprehenderlas. Debe destacarse que la posición de recogida siempre sitúa la mano de tal modo que la yema de los dedos extendidos forma parte del plano meridional horizontal de la esfera a recoger.

Las consignas obtenidas durante el experimento (ver Tabla 5.2) permiten recoger las esferas, y moverlas con una aceleración máxima igual a la del brazo TX-90. Si se realiza la cinemática directa de la mano con los valores obtenidos, se obtiene que las yemas de los cuatro dedos forman una esfera que tiene un radio menor que la esfera recogida. Esto se debe a que actualmente no existe un control de fuerza de la mano, y por lo tanto la consigna de posición debe generar unos puntos interiores a la esfera para poder ejercer fuerza sobre ella.

Diámetro esfera [mm]	160	150	110
Anular articulación 0 [º]	0	0	10
Anular articulación 1 [º]	25	30	35
Anular articulación 2 [º]	65	65	65
Anular articulación 3 [º]	20	20	20
Medio articulación 0 [º]	0	0	0
Medio articulación 1 [º]	20	25	30
Medio articulación 2 [º]	65	65	65

Medio articulación 3 [º]	20	20	20
Índice articulación 0 [º]	0	0	-10
Índice articulación 1 [º]	25	30	45
Índice articulación 2 [º]	65	65	65
Índice articulación 3 [º]	20	20	20
Pulgar articulación 0 [º]	0	0	0
Pulgar articulación 1 [º]	20	20	20
Pulgar articulación 2 [º]	10	10	10
Pulgar articulación 3 [º]	30	30	35

Tabla 5.2 – Valores de las articulaciones de MA-I para tres esferas.

Una vez obtenidos estos valores se ha realizado una regresión lineal para aproximar el valor de las articulaciones a partir del diámetro de la esfera. Las articulaciones que varían en función del diámetro se rigen por el grupo de ecuaciones (Ec. 5.4) donde A0 corresponde a la articulación 0 del dedo anular, y I1, por ejemplo, corresponde a la articulación 1 del dedo índice (consultar Figura 3.2 de la página 12).

$$
\begin{aligned}
A0 &= -0.21 \cdot d + 33 \\
A1 &= -0.18 \cdot d + 55 \\
M1 &= -0.18 \cdot d + 50 \\
I0 &= 0.21 \cdot d - 33 \\
I1 &= -0.39 \cdot d + 88 \\
P3 &= -0.11 \cdot d + 47
\end{aligned}
\qquad \text{(Ec. 5.4)}
$$

Otra consideración experimental particular de ESFERA es la definición de la zona de trabajo. Este entorno está descrito en el Apartado 5.1, y corresponde a una superficie plana y rectangular de 900x400mm situada sobre la mesa que sirve de soporte para el brazo robot.

Resulta evidente que la superficie de trabajo se encuentra limitada por la zona de mesa que se observa en la imagen menos la zona de seguridad establecida frente a posibles

colisiones del MR con la base del robot. Sin embargo, existen dos factores que obligan a disminuir esta zona para efectuar la localización de una esfera. El primero es el diámetro de la esfera, y el segundo el ángulo de apertura de la cámara.

En la Figura 5.6 mostrada a continuación se puede observar la sección que produce un plano vertical que pasa por el centro de la esfera y la cámara y es perpendicular a un plano vertical que pasa por un segmento del contorno de la imagen. Siendo a el ángulo de apertura de la cámara, y situando la esfera en el límite de su representación completa en la imagen.

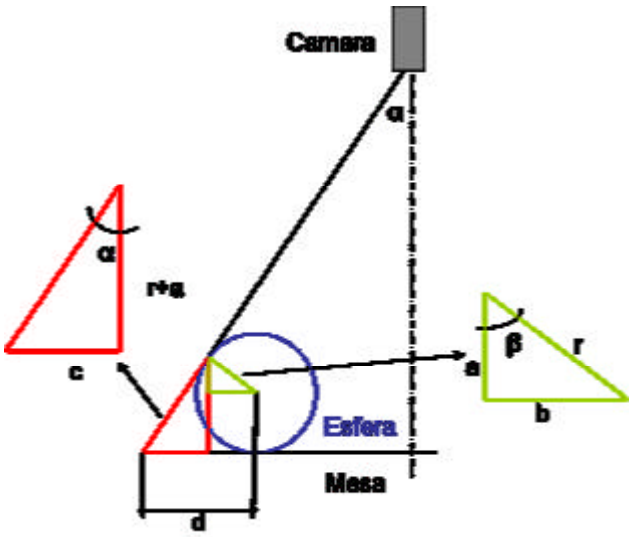


Figura 5.6 – Esquema para la determinación de la superficie de trabajo.

Como se puede comprobar en la imagen anterior, el centro de la esfera debe situarse a una distancia (llamada d) por dentro del borde de la imagen si no se quiere que quede recortada. El valor de d se obtiene según la Ec. 5.5 a partir del ángulo de apertura a y el radio de la esfera (r).

$$d = r \cdot (\tan(a) \cdot (1 + \cdot \cos(90 - a)) + \sin(90 - a)) \qquad \text{(Ec. 5.5)}$$

El ángulo de apertura de la cámara se puede obtener a partir de la Ec. 5.6. Siendo h la altura de la cámara respecto la mesa, y dx la distancia del centro de la imagen a un segmento del contorno de la imagen.

$$a = \arctan(dx / h) \qquad \text{(Ec. 5.6)}$$

A partir de los valores h, dv y dh (ver Figura 5.7) obtenidos experimentalmente, se calculan los ángulos de apertura vertical (a) y horizontal (ß), que son de 15.34 y 20.10 grados

respectivamente. Según los valores de apertura se obtiene una distancia mínima del centro de la esfera al borde de la imagen para los segmentos horizontales del contorno de 114.5mm y de 104.9mm para los verticales.

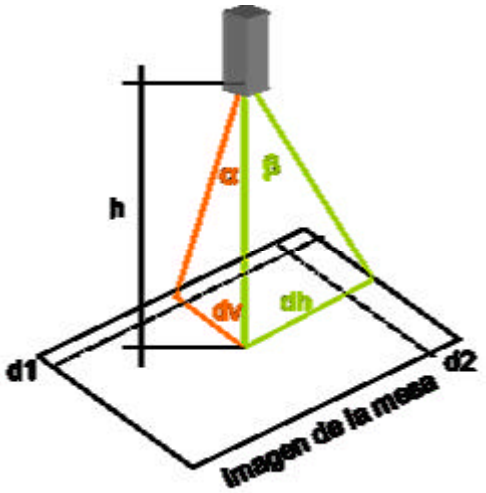


Figura 5.7 – Ángulos de apertura de la cámara.

Finalmente, para evitar una excesiva sensibilidad a las perturbaciones, se fija la distancia a la que se puede situar el centro de una esfera de los márgenes superior e izquierdo de la imagen para su recogida de 150mm. La distancia establecida para los márgenes inferior y derecho son superiores debido a la cercanía de las bases de los robots del laboratorio (ver Figura 4.5 de la página 37) redundando así en la política de seguridad frente a las colisiones.

Teniendo en cuenta las especificaciones descritas anteriormente se obtiene una configuración de la zona de trabajo de ESFERA respecto a la imagen y la mesa como la que se muestra en la Figura 5.8.

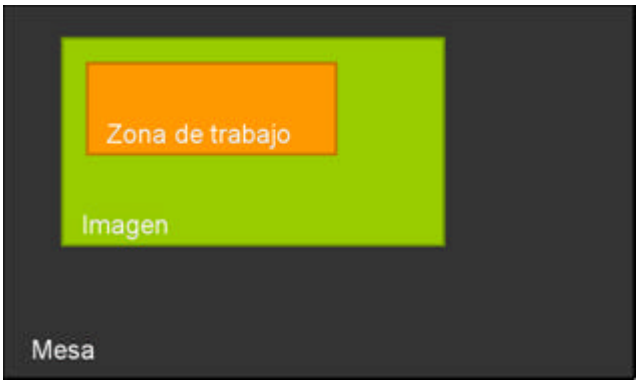


Figura 5.8 – Zona de trabajo de ESFERA.

A continuación se describen los pasos que se deben seguir para el uso de la tarea ESFERA:

1. Ejecutar la interfase VMR.

2. Alimentar la cámara y comprobar que los parámetros de la imagen están correctamente introducidos en el panel de calibración del VMR. En la actualidad los valores por defecto obtenidos experimentalmente son los siguientes: escala de 3.436mm/píxel, ángulo de rotación de 1º y coordenadas del origen de la imagen (989, 478, -478).

3. Ejecutar el programa MAIQNX6 desde un terminal del ordenador EL-FEO. Alimentar la etapa de potencia. Pulsar el botón de inicialización de la interfase. Consultar la metodología de inicialización de la mano en el apartado 3.1.3 de la página 16.

4. Pulsar el botón Socket Server de la interfase MAIQNX6 para ejecutar el MAIQNX6 Server. Comprobar la dirección IP del equipo, y el puerto del servidor. Pulsar el botón Start Receiving para esperar una conexión.

5. Encender el armario de control CS8C. Comprobar la dirección IP del armario, y que los parámetros del socket creado ServerTele son los adecuados: valor plazo de activación de error 0 y carácter fin de cadena 64 (comprobar también el puerto del socket). Consultar los protocolos de puesta en marcha del armario descritos en el apartado 3.2.5 de la página 28, así como la información de los sockets de tipo servidor del apartado 3.2.2 en la página 22.

6. Cargar la aplicación VMR VAL3. Consultar los protocolos de ejecución de una aplicación en modo local del apartado 3.2.5 de la página 28.

7. Comprobar que las direcciones IP y los puertos de la interfase VMR son los adecuados. Ejecutar la sentencia “ping –t dirección IP del CS8C” y ““ping –t dirección IP del EL-FEO” en dos ventanas de comandos para comprobar la conexión con el armario y el ordenador de la mano, y mantenerla activa. Conectar el VMR con la mano y el robot.

8. Realizar el Home del sistema VMR. Nótese que esta etapa implica movimiento.

9. Definir la posición de entrega de la Tarea3.

10. Situar una esfera en la zona de trabajo y ejecutar ESFERA.

A continuación se muestran unas imágenes tomadas durante la experimentación real de ESFERA. En el transcurso de la tarea se realizan los siguientes pasos: posicionamiento de la esfera en el entorno de trabajo (1), Home (2), posición de seguridad (3), posición de recogida (4), aprehensión de la esfera (5), posición de seguridad (6), posición de entrega (7), extensión de la mano (8) y entrega de la esfera (9), según la Figura 5.9.

Figura 5.9 – Ejecución de ESFERA.

6. Evaluación de Impacto Ambiental

El proyecto de Integración de visión por computador en un sistema Mano-mecánica / Robot es, fundamentalmente, un proyecto de diseño informático. La implantación del proyecto se ha pensado y realizado para el IOC, por lo que la legislatura aplicable, es la europea, la española y la catalana.

En el diseño y realización de proyectos de carácter informático no hay involucrado un impacto directo sobre el medio [ETSEIB, 1999, p. 17]. No obstante, en este caso, también se ha implantado un sistema físico, por lo que su impacto en el medio será contemplado en la evaluación.

Los efectos negativos del proyecto sobre el medio durante a su fase de diseño son principalmente los siguientes:

- Emisiones debidas al consumo de energía eléctrica. Se estima que se han consumido una media de 30w (mano), 140w (dos PC + cámara), 1200w (robot), 200w (iluminación), 800w (climatización), durante ocho horas al día en seis meses, lo que corresponde a una energía de 3412.8kwh.

- Consumo de papel, tinta, piezas de poliestireno expandido.

- Emisiones debidas al transporte diario al IOC (medio utilizado: metro).

Los efectos negativos del proyecto sobre el medio durante a su fase de explotación son principalmente los siguientes:

- Emisiones debidas al consumo de energía eléctrica. Teniendo el sistema completo una potencia aproximada de 2370w (mano + dos PC + cámara + robot + iluminación + climatización).

- Consumo de papel, y tinta, debido al uso en investigación del sistema.

El efecto negativo del proyecto sobre el medio durante su fase de desmantelamiento se debe a los residuos del sistema, compuestos principalmente de metales y plásticos.

Para paliar los efectos adversos del proyecto en el medio, se proponen las siguientes medidas correctoras:

- Utilización de PC's con sistemas de refrigeración eficientes (para evitar el uso de la climatización de la sala con ese fin), activación de desconexiones automáticas en las pantallas, la cámara, etc.

- Reciclaje de los residuos generados por el uso del sistema (papel, tinta).

- Uso de iluminación eficiente (por ejemplo fluorescentes) en el laboratorio.

- Gestión medioambiental de los residuos generados durante el desmantelamiento del sistema.

Finalmente, cabe destacar, que el proyecto puede tener un impacto positivo sobre la sociedad, debido a la contribución que hace en el campo de la automatización de tareas, pudiendo aportar mayor eficiencia en ciertos procesos.

7. Presupuesto

El objetivo del proyecto es integrar visión por computador en el sistema MR del IOC. Para establecer el presupuesto adecuado a la tipología del proyecto es necesario conocer las etapas que se han desarrollado en su consecución.

Fundamentalmente se han realizado las siguientes fases: Fase creativa (o de diseño) en la que se traducen los objetivos en un proyecto definido. Fase de construcción en la que se transforma el proyecto en una realidad física. Y finalmente, fase de explotación en la que se obtienen los frutos del proyecto.

Por lo tanto, se ha considerado que el presupuesto debe corresponder a un estudio de consultoría (fase de diseño), de implantación (fase de construcción) y de producción (fase de explotación) del sistema VMR en el IOC. Nótese que no se contemplan en el presupuesto los costes relativos a la implantación de la estructura existente antes de la realización del proyecto.

El presupuesto se ha estructurado en dos grandes apartados de costes:

- Coste de los recursos humanos. Constituye la partida más importante del total del presupuesto. Se cuantifican los costes relacionados con los trabajos realizados por los ingenieros y el personal administrativo implicados en el proyecto.

- Coste de los recursos materiales. Se cuantifica el coste de los productos que ha adquirido el IOC para la realización del proyecto.

El cálculo de los costes de los recursos humanos se ha estructurado teniendo en cuenta tres grandes aspectos:

- Las fases del proyecto en su proceso de elaboración.

- Las tarifas horarias de los diferentes profesionales que han intervenido en la realización del proyecto.

- La dedicación horaria de los profesionales en cada fase.

A continuación se recoge en la Tabla 7.1 los costes de los recursos humanos según las fases del proyecto y la categoría de los profesionales que han intervenido en cada una de ellas.

Fase del proyecto	Profesional	Horas	Tarifa [€/h]	Coste [€]
Diseño	Consultor experto	30	150	4500
	Ingeniero Sénior	70	70	4900
	Ingeniero Júnior	250	35	8750
Construcción	Consultor experto	40	150	6000
	Ingeniero Sénior	220	70	15400
	Ingeniero Júnior	660	35	23100
	Administrativo	10	30	300
Explotación	Consultor experto	10	150	1500
	Ingeniero Sénior	30	70	2100
	Ingeniero Júnior	140	35	4900
			Total [€]	71450

Tabla 7.1 – Tabla de costes de los recursos humanos.

En la fase de diseño y explotación se ha considerado el tiempo de trabajo del director del proyecto, el Sr. Raúl Suarez, que ha ejercido las funciones de consultor experto, y del Sr.Carlos Rosales como Ingeniero Sénior, por su colaboración en esta fase. Asimismo se ha considerado la categoría de Ingeniero Junior para el proyectista Miguel Quesada.

Durante la realización de la fase de construcción se han desarrollado los siguientes trabajos: compra de material, implantación del sistema, desarrollo y verificación del software, y redacción de la documentación del proyecto. En la compra de material ha intervenido el personal administrativo del IOC. El trabajo de implantación del sistema y desarrollo del software ha sido realizado por los profesionales de la fase de diseño en colaboración con los señores Patrick Grosh y Leopold Palomo, ambos con categoría de Ingeniero Sénior.

El cálculo de los costes de los recursos materiales se ha desglosado en las siguientes partidas:

- Compras. Adquisición de la tarjeta Angelo RTV-24, figuras de poliestireno expandido.
- Consumo eléctrico (ordenadores, TX-90, MA-I, climatización, etc.).
- Material de oficina (papel de impresora, fotocopias, cartuchos de tinta, CD's, etc. Incluye la creación de la documentación del proyecto)
- Comunicaciones (teléfono, FAX, internet).
- Desplazamientos a proveedores.

En la Tabla 7.2 se muestran los costes de los recursos materiales según las partidas descritas previamente.

Concepto	Facturación [€]
Compras	
Angelo RTV-24	210
Figuras porex	30
Consumo eléctrico	440
Material de oficina	300
Comunicaciones	250
Desplazamientos	40
Total	1270

Tabla 7.2 – Tabla de costes de los recursos materiales.

Finalmente, el coste total del proyecto se obtiene de la suma de los costes parciales de recursos humanos y de materiales. El resultado se muestra, a continuación, en la Tabla 7.3.

Concepto	Importe [€]
Recursos humanos	71450
Recursos materiales	1270
Total	72720
Beneficio industrial (20%)	14544
Total + Beneficio industrial	87264
IVA (16%)	13962
Total + IVA	101226

Tabla 7.3 – Presupuesto del proyecto.

Finalmente, se obtiene, tras las imposiciones oportunas, que el presupuesto total del proyecto es de 101226 €.

Conclusiones

La conclusión principal del proyecto es que la integración de visión por computador en el sistema MR se ha realizado con éxito, cumpliendo con todos los objetivos establecidos en la fase de diseño.

Actualmente, el sistema VMR sirve como plataforma de pruebas experimentales para la localización, aprehensión y manipulación de objetos, como se ha demostrado en la aplicación real desarrollada.

El software del sistema VMR permite la ampliación de las tareas que realiza actualmente, así como la creación de otras nuevas. Además, el VMR se puede ejecutar en diversos sistemas operativos, lo que permite su uso en todos los computadores del IOC. Estas características hacen del sistema desarrollado una herramienta flexible y potente para los investigadores del instituto.

Se ha equipado al laboratorio del IOC con un software que permite el comando remoto del sistema MR desde un entorno único y de forma simultánea. Con lo que se cubre una necesidad del IOC.

Finalmente, y en clave más general, se ha aportado un ejemplo práctico de la capacidad de trabajo de los sistemas robotizados con visión por computador. El proyecto demuestra que este tipo de sistemas pueden ser utilizados como herramientas útiles en la industria para realizar tareas complejas.

A modo de recomendación, se sugiere una futura integración en el sistema VMR de herramientas que permitan la reconstrucción en 3D de escenas y de análisis de movimiento. También puede ser de interés, introducir en el sistema VMR algoritmos y técnicas de aprehensión a partir de modelos tridimensionales de objetos. Así como sistemas de control de colisiones.

Agradecimientos

Quiero agradecer al Doctor Raúl Suarez Feijóo por haberme dado la posibilidad de desarrollar un proyecto tan interesante como este.

Gracias a Carlos Rosales Gallegos por el seguimiento y el apoyo en el desarrollo de este proyecto. Su profunda dedicación y motivación han hecho posible la consecución de los objetivos.

Quiero agradecer también, las aportaciones e indicaciones de Patrick Grosch, Emmanuel Nuno, Leopold Palomo y Adolfo Rodríguez.

Por último, me gustaría agradecer por su ayuda y colaboración a todo el grupo de ingenieros del laboratorio de robótica del IOC.

Bibliografía

Referencias bibliográficas

COMER, D. E. *Redes globales de información con internet y TCP/IP*. México, Prentice Hall, 2000.

CRAIG J., *Introduction to Robotics mechanics and control*. Massachusetts, Addison-Wesley Publishing Company Inc., 2ª Ed., 1989.

ESCOLA TÉCNICA SUPERIOR D'ENGINYERIA INDUSTRIAL DE BARCELONA (ETSEIB). *GUIA DE CONTENIDOS AMBIENTALES DE LOS PROYECTOS FIN DE CARRERA.* Barcelona, 1999.

LANGSCH, F. *Diseño e Implementación de un Sistema de Tiempo Real para el Control de una Mano Mecánica.* Proyecto Final de Carrera. Febrero 2005.

STÄUBLI. *Manual de instrucciones. Armario de control CS8C.* Julio, 2004a. [CD].

STÄUBLI. *Manual de instrucciones. Brazo Familia TX Serie 90.* Junio, 2004b. [CD].

SUAREZ, R., GROSCH, P. *Mano Mecánica MA-I*. XXIV Jornadas de Automática, CEA-IFAC. León, 10-12 de Septiembre, 2003.

Bibliografía complementaria

En primer lugar se lista la bibliografía básica para la realización del proyecto.

ADLINK. *RTV Series Multi-Channel Real-Time vides Frame Grabber Series. User's Manual.* Octubre, 2004. [CD].

SONY. *Color Camera Module. Technical Manual.*2005.

STÄUBLI. *Manual de referencia VAL3. Versión 5.0.* Septiembre, 2004. [CD].

También se muestra la bibliografía de consulta.

ESCOLA TÉCNICA SUPERIOR D'ENGINYERIA INDUSTRIAL DE BARCELONA. *Sistema Internacional de Unidades.* [http://edison.upc.es/units/SIcas.html, 14 Agosto 2007].

GOTTFRIED, B. *Programación en C.* Madrid, McGraw-Hill, 1997.

STROUSTRUP, B. *El Lenguaje de programación C++*. Argentina, Addison Wesley Turpial cop. 1998.

SUAREZ, R., GROSCH, P. *Modelado y control de una mano mecánica*. 1ª Jornada de Recerca en Automàtica, Visió i Robòtica, AVR'2004. Barcelona, 3-4 de Febrero, 2004. p. 299-306.