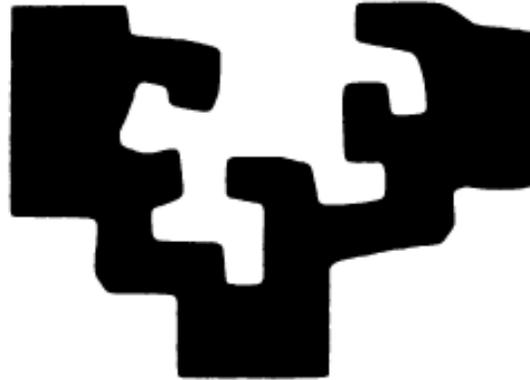


eman ta zabal zazu



universidad
del país vasco

euskal herriko
unibertsitatea

Facultad de Informática Informatika Fakultatea

TITULACIÓN: Ingeniería Informática

Alumno: D. Alberto García Marín

Director: D. Julio Abascal González

Sistema de entrenamiento para un brazo
articulado portátil

Proyecto Fin de Carrera, Julio 2012

© 2012 Alberto García



RESUMEN

El proyecto está enmarcado dentro del ámbito de la robótica enfocada a la rehabilitación, asistencia y compensación funcional de personas con discapacidades motoras. Se ha querido desarrollar un sistema de entrenamiento en el que el usuario final pueda decidir qué interfaces y configuraciones se adaptan mejor a sus habilidades para controlar el brazo articulado iARM. Como resultado del proyecto se presenta el siguiente documento.

The project can be classified in the field of robotics focused on rehabilitation, assistance and functional compensation of persons with motor disabilities. It has been wanted to develop a training system in which the end user can decide which interfaces and configurations are best suited to their abilities to control the iARM. As a result of the project is provided this document.

PALABRAS CLAVE

iARM, interfaz, Qt, feedback, CAN-bus, joystick





INDICE

1. Documento de objetivos del proyecto	1
1.1. Título.....	1
1.2. Descripción	1
1.3. Objetivos del proyecto fin de carrera	2
1.4. Diagrama E.D.T.	4
1.5. Fases del proyecto.....	5
1.6. Gestión y planificación	6
1.6.1. Programación temporal	6
1.6.2. Diagrama de Gantt	6
1.6.3. Estimación de tiempos de las tareas	8
2. Estado del Arte.....	9
2.1. Robots de asistencia.....	9
2.1.1. Tipos de robots de asistencia	11
2.1.2. IArm	17
2.2. Interfaces para personas con discapacidades.....	19
2.2.1. Interfaces para el iArm.....	20
3. Hardware y software usado en el proyecto final de carrera.....	23
3.1. Hardware	23
3.1.1. El robot iArm	23
3.1.2. USB-CAN interface.....	25
3.1.3. <i>Joystick</i>	26
3.2. Software.....	27
3.2.1. C++ en el proyecto	27
3.2.2. Compilador MinGW	27
3.2.3. Qt Creator como entorno de desarrollo	28
3.2.4. Qt, la librería	28
3.2.5. API iARM	29
3.2.6. SDL.....	29
4. Desarrollo de un sistema de entrenamiento.....	31
4.1. Introducción	31



4.2.	Modelo de casos de uso	32
4.2.1.	Desplegar	33
4.2.2.	Plegar	34
4.2.3.	Cambiar Velocidad	34
4.2.4.	Mover (Cartesiano)	35
4.2.5.	Mover (Orientación de la pinza).....	36
4.2.6.	Mover (Articulaciones)	36
4.2.7.	Mover (Elevador)	37
4.2.8.	Mover (Posición de Inicio)	37
4.2.9.	Abrir/Cerrar Pinza	38
4.2.10.	Iniciar Robot.....	38
4.3.	Sistema de entrenamiento. Parte I: Uso de una interfaz gráfica de usuario.....	39
4.3.1.	Diagrama de secuencia	41
4.4.	Sistema de entrenamiento. Parte II: Uso del Joystick.....	42
4.4.1.	Diagrama de secuencia	44
5.	Revisión del documento de objetivos del proyecto.....	47
5.1.	Revisión de los objetivos y planificación del proyecto.....	47
5.1.1.	Análisis de las horas realizadas en el proyecto	48
5.1.2.	Historia del proyecto	50
6.	Conclusiones y líneas de trabajo futuras	51
6.1.	Conclusiones	51
6.2.	Líneas de trabajo futuras	51
7.	Bibliografía	55



INDICE DE FIGURAS

Fig. 1 Estructura Desagregada del trabajo.....	4
Fig. 2 Diagrama de Gantt.....	7
Fig. 3 Entorno del usuario final.....	10
Fig. 4 PUMA 250.....	12
Fig. 5 Handy 1.....	12
Fig. 6 Raid.....	14
Fig. 7 Sistema ProVar.....	14
Fig. 8 MoVAR.....	15
Fig. 9 MoVAID.....	15
Fig. 10 Raptor.....	16
Fig. 11 ASIBOT.....	17
Fig. 12 iArm.....	18
Fig. 13 Pinza del iARM.....	19
Fig. 14 Dispositivo para controlar el iArm.....	20
Fig. 15 Selección de objetos usando la interfaz propuesta por la universidad de Massachussets Lowell.....	21
Fig. 16 Esquema de comunicación iArm.....	23
Fig. 17 Área de trabajo del iArm.....	24
Fig. 18 USD-CAN interface.....	25
Fig. 19 <i>Joystick</i> . Force 3D PRO.....	26
Fig. 20 Modelo de casos de uso.....	32
Fig. 21 Sistema de entrenamiento (Parte I). Interfaz gráfica.....	39
Fig. 22 Recuadro que simula un 'touchpad'.....	40
Fig. 23 Diagrama de secuencia. Sistema de entrenamiento (Parte I).....	42
Fig. 24 Interfaz para la configuración del joystick.....	43
Fig. 25 Diagrama de secuencia. Sistema de entrenamiento (Parte II).....	45
Fig. 26 Grafico con la comparación de horas de las tareas principales.....	48



INDICE DE TABLAS

Tabla 1 Tiempos de las tareas	8
Tabla 2 Caso de uso: Desplegar	33
Tabla 3 Caso de uso: Plegar	34
Tabla 4 Caso de uso: Cambiar Velocidad	35
Tabla 5 Caso de uso: Mover (Cartesiano)	35
Tabla 6 Caso de uso: Mover (Orientación de pinza)	36
Tabla 7 Caso de uso: Mover (Articulaciones).....	37
Tabla 8 Caso de uso: Mover (Elevador).....	37
Tabla 9 Caso de uso: Mover(Posición de Inicio)	37
Tabla 10 Caso de uso: Abrir/Cerrar Pinza	38
Tabla 11 Caso de uso: Iniciar Robot	38
Tabla 12 Tabla detallada con la comparación de horas de cada tarea	49



1. Documento de objetivos del proyecto

1.1. Título

“Sistema de entrenamiento para un brazo articulado portátil”

1.2. Descripción

El proyecto se puede enmarcar dentro del ámbito de la robótica enfocada a la rehabilitación, asistencia y compensación funcional de personas con discapacidades motoras. Lo que se quiere conseguir en este proyecto es dotar a un brazo articulado portátil de un sistema de entrenamiento en el que el usuario final pueda elegir la configuración final que más se ajuste a sus habilidades para dirigir el robot.

El brazo articulado del que se dispone es un iARM (Intelligent Arm Robot Manipulator). Se trata de un brazo articulado diseñado para ir montado en una silla de ruedas y ayudar al usuario a manipular objetos para realizar tareas cotidianas de la vida real. Este robot puede ser controlado mediante el uso de un teclado especial. Pero esta interfaz no ofrece una gran facilidad para mover y controlar el brazo articulado de manera sencilla y requiere varias iteraciones para mover las articulaciones del robot para poder realizar los movimientos del brazo articulado. Además de esto, no ofrece al usuario la posibilidad de tener una configuración personalizable.

Por tanto, el reto del proyecto es realizar un sistema de entrenamiento mediante el diseño de interfaces avanzadas y accesibles para el control de un brazo articulado portátil instalado en una silla de ruedas eléctrica.

A la hora de realizar la interfaz avanzada se van a tener en cuenta los distintos dispositivos de interacción existentes (joystick, mini-teclado, voz, pantalla táctil...). Para ello habrá que tener en cuenta el material del que se dispone en el laboratorio de interacción persona-computador para necesidades especiales.

Una vez seleccionados los dispositivos de interacción se quiere hacer una interfaz avanzada para tener un control más accesible y fácil del brazo



articulado. Se tendrá en cuenta que el usuario final de la interfaz serán personas con discapacidad motora y, por lo tanto, se deberán analizar las posibles restricciones.

Este último punto es una de las características más importantes del proyecto y que hay que prestar mayor atención, ya que no se trata solo de hacer una interfaz accesible para el uso del brazo articulado. Sabemos que nuestro usuario final tiene algún tipo de profunda discapacidad motora, pero todos los usuarios no tendrán las mismas habilidades. Es por ello que queremos dotar de un sistema de entrenamiento donde el usuario final pueda decidir cuál es la configuración final que más se ajusta a sus habilidades.

1.3. Objetivos del proyecto fin de carrera

Los objetivos que se van a alcanzar en este proyecto final de carrera son los siguientes:

- Estudio del brazo articulado iARM.
 - Pruebas, configuración e instalación del robot.
 - Análisis de la funcionalidad actual del brazo articulado.
- Estudio de la interacción persona-robot
 - Análisis sobre brazos articulados para personas con discapacidad.
- Estudio de los dispositivos de interacción avanzada.
 - Análisis de las distintas posibilidades existentes para el uso de dispositivos de interacción (*Joystick*, mini-teclado, voz, pantalla táctil...)
- Desarrollo de una interfaz para el brazo articulado.
 - Implementación de una aplicación eficiente y sencilla, acorde a las características del robot.



- Creación de interfaces de usuario con alto grado de usabilidad para el control del robot.
- Evaluación de la usabilidad de aplicación.
 - Realizar un análisis para comprobar el grado de usabilidad

El objetivo principal es desarrollar una interfaz avanzada y accesible para el control de un brazo articulado portátil instalado en una silla de ruedas. Se ha dividido el proyecto en distintas fases.

La primera fase será de formación, prestando atención a la documentación sobre el iArm, literatura sobre interacción persona-robot (especialmente brazos articulados para personas con discapacidad motora) y estudio de dispositivos de interacción. La segunda fase será la puesta en marcha e instalación del *hardware* y el *software*. Se instalarán los *drivers* del brazo articulado y los dispositivos de interacción, así como el entorno de desarrollo y librerías a utilizar. La tercera fase, se dedicará al desarrollo de la interfaz combinada avanzada y realización de pruebas sobre el brazo articulado.

Los objetivos que se han comentado son los que se plantearon al inicio del proyecto, aunque variaron a lo largo de este. Tal como se ha comentado anteriormente se trata de un proyecto en el que los usuarios finales son personas con gran discapacidad motora, con diversas habilidades, por lo que es muy importante que la interfaz sea configurable. Por ello se fijó también como objetivo desarrollar un sistema de entrenamiento en el que el usuario final pudiese elegir la configuración final que más se ajustase a sus habilidades. Además se propuso hacer otra interfaz que complementara el sistema de entrenamiento, en este caso una interfaz gráfica en la que el usuario final pudiese interactuar con ella mediante una pantalla táctil o mediante el uso del ratón.



1.4. Diagrama E.D.T.

EDT o Estructura Desagregada del Trabajo sirve para separar el proyecto en un conjunto de componentes principales que a su vez se aíslan en componentes cada vez más pequeños. Por tanto la planificación del proyecto quedaría del siguiente modo:

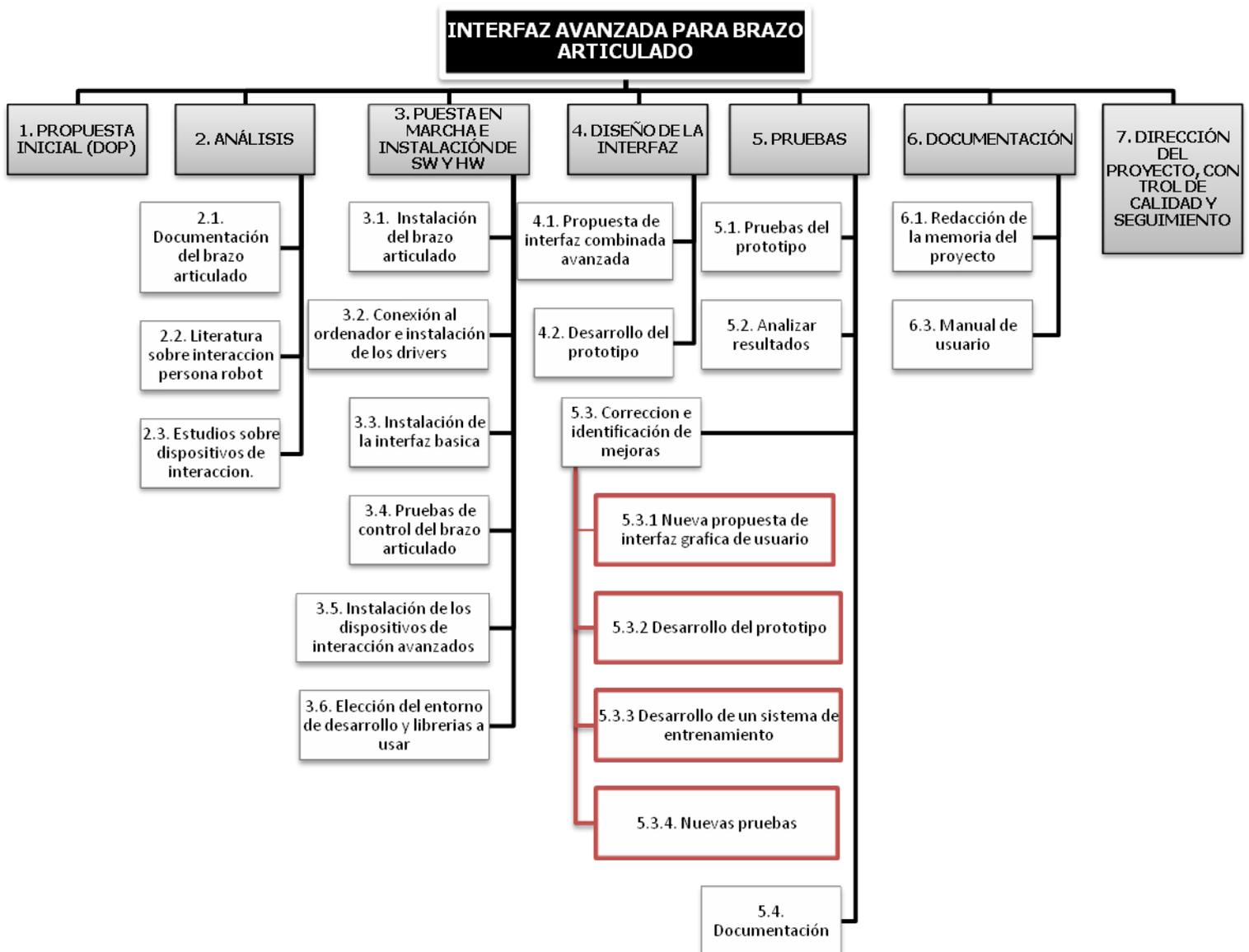


Fig. 1 Estructura Desagregada del trabajo



Los componentes en rojo pertenecen a la replanificación durante el proyecto que incluyen el diseño de una nueva interfaz. En este caso se trata de una interfaz gráfica de usuario y el desarrollo de un sistema de entrenamiento.

1.5. Fases del proyecto

El listado de las fases y tareas del proyecto son las siguientes:

- 0 Interfaz avanzada para brazo articulado portátil**
- 1 Propuesta inicial (Documento de objetivos del proyecto)**
- 2 Análisis**
 - 2.1 Documentación del brazo articulado
 - 2.2 Literatura sobre interacción persona robot
 - 2.3 Estudio sobre los dispositivos de interacción
- 3 Puesta en marcha e instalación del *hardware* y *software***
 - 3.1 Instalación del brazo articulado
 - 3.2 Conexión al ordenador e instalación de los *drivers*
 - 3.3 Instalación de la interfaz básica
 - 3.4 Pruebas de control del brazo articulado
 - 3.5 Instalación de los dispositivos de interacción avanzados
 - 3.6. Elección del entorno de desarrollo y librerías a usar
- 4 Diseño de la interfaz**
 - 4.1 Propuesta de interfaz combinada avanzada
 - 4.2 Desarrollo del prototipo
- 5 Pruebas**
 - 5.1 Pruebas del prototipo
 - 5.2 Análisis de los resultados
 - 5.3 Corrección e identificación de mejoras
 - 5.3.1 Nueva propuesta de interfaz gráfica de usuario
 - 5.3.2 Desarrollo del prototipo
 - 5.3.3 Desarrollo de un sistema de entrenamiento
 - 5.3.4. Nuevas pruebas
 - 5.4 Documentación



6 Documentación

6.1 Redacción de la memoria del proyecto

6.2 Manual de usuario

7 Dirección del proyecto, control de calidad y seguimiento

1.6. Gestión y planificación

En este apartado se considerará la parte de planificación y gestión del proyecto. Se muestra un estudio de la planificación temporal, que contiene por un lado el diagrama de Gantt y, por otro lado, una estimación de tiempos de las tareas tras las replanificación final.

1.6.1. Programación temporal

El proyecto estaba planificado para iniciarse en el año 2011 con el comienzo del segundo cuatrimestre universitario y finalizar en julio del mismo año.

Fecha de inicio: 24 de enero de 2011.

Fecha de finalización: 13 de julio de 2011.

Miembros del proyecto: Alberto García Marín.

Días laborables: Se trabajará de lunes a viernes.

Jornadas de trabajo:

DIARIA: 3 horas de lunes a viernes intensificándolo a 6 horas en el mes final.

Estimación total en horas de trabajo: 387 horas.

1.6.2. Diagrama de Gantt

A continuación se puede ver el diagrama de Gantt de la planificación del proyecto.



1.6.3. Estimación de tiempos de las tareas

TAREA PRINCIPAL	SUBTAREA	DURACIÓN TAREA
1. Propuesta inicial (Documento de objetivos del proyecto)		12 horas
2. Análisis (24h)	Documentación del brazo articulado	6 horas
	Literatura sobre interacción persona robot	14 horas
	Estudio sobre los dispositivos de interacción	4 horas
	Instalación del brazo articulado	3 horas
3. Puesta en marcha e instalación del hardware y software (24h)	Conexión al ordenador e instalación de los drivers	3 horas
	Instalación de la interfaz básica	6 horas
	Pruebas de control del brazo articulado	2 horas
	Instalación de los dispositivos de interacción avanzados	4 horas
	Elección del entorno de desarrollo y librerías a usar	6 horas
4. Diseño de la interfaz (80h)	Propuesta de interfaz combinada avanzada	30horas
	Desarrollo del prototipo	50 horas
	Pruebas del prototipo	15 horas
5. Pruebas (135h)	Analizar resultados	10 horas
	<u>Corrección e identificación de mejoras</u>	
	Nueva propuesta de interfaz gráfica de usuario	14 horas
	Desarrollo del prototipo	40 horas
	Desarrollo de un sistema de entrenamiento	32 horas
	Nuevas pruebas	16 horas
6. Documentación (90h)	Documentación	8 horas
	Redacción de la memoria del proyecto	80 horas
	Manual de usuario	10 horas
7. Dirección del proyecto, control de calidad y seguimiento		22 horas
		387 horas

Tabla 1 Tiempos de las tareas



2. Estado del Arte

Desde hace décadas la implantación de la Robótica en la industria es una realidad. Hoy en día, sin embargo, el reto de la Robótica es su incorporación a la vida cotidiana, fuera de las aplicaciones industriales tradicionales. Para alcanzar este objetivo se requieren robots capaces de realizar tareas de forma autónoma, manipular objetos, moverse en entornos diseñados para humanos, interactuar y colaborar con personas y que simultáneamente, cumplan unos criterios de robustez operacional y seguridad física elevados.

Una de las áreas de la Robótica en la que los investigadores están trabajando últimamente es en robots de asistencia para personas con discapacidad, de manera que puedan lograr cierta independencia en la realización de tareas de la vida cotidiana. En los últimos años, se ha avanzado mucho en los dispositivos robóticos pero no tanto en el desarrollo de interfaces que ayuden a las personas con discapacidades.

2.1. Robots de asistencia

Para llevar a cabo tareas de la vida cotidiana como manipular objetos, comer, beber, abrir puertas u otras actividades, las personas con discapacidad necesitan ayuda externa. Los dispositivos robóticos pueden servir como asistencia en estas situaciones. Estos dispositivos pueden hacer la función del brazo y la mano, y pueden ser controladas mediante una interfaz accesible a la persona discapacitada.

En la Fig.3 podemos ver un esquema del entorno en el que se mueven las personas discapacitadas. Para poder manipular objetos del entorno estas personas disponen de dispositivos de asistencia tales como controles remotos para abrir o cerrar puertas, encender o apagar luces o la televisión, etc. Los dispositivos robóticos forman parte de una categoría especial de los dispositivos de asistencia. Esta categoría es más compleja que los dispositivos de asistencia usuales por la complejidad del aparato: controlar un brazo articulado de seis grados de libertad es mucho más complejo que por ejemplo activar una puerta automática.



Además, el usuario final no solo quiere controlar objetos en entornos estructurados (puertas, objetos puestos en lugares predeterminados...), sino que quiere ser capaz de manipular objetos en entornos no estructurados (objetos puestos en lugares aleatorios). Este objetivo es mucho más complejo que operar con dispositivos de asistencia.

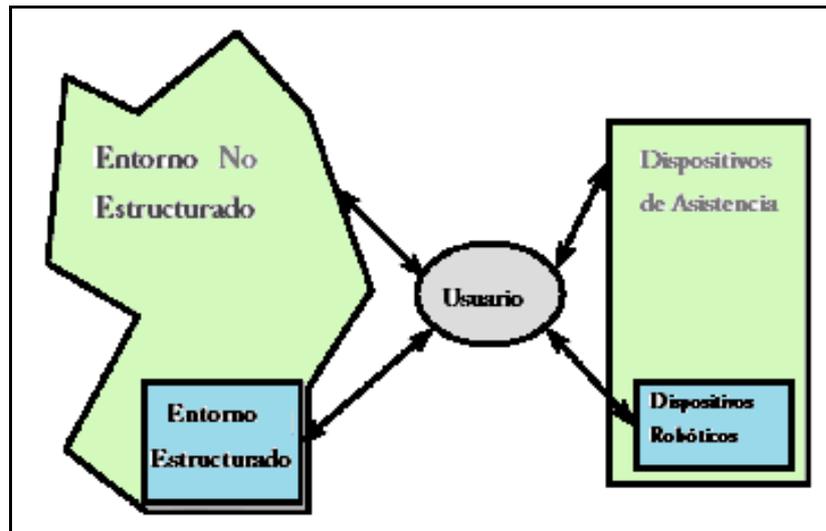


Fig. 3 Entorno del usuario final

Se pueden seguir dos enfoques distintos para operar con el dispositivo robótico en un entorno no estructurado. El primero es utilizar distintos sensores para recibir información del entorno no estructurado e identificar los objetos. Esta opción puede dar muy buenos resultados en un futuro cercano, pero actualmente la complejidad de sistema puede resultar muy cara. El segundo enfoque deja en manos del usuario final la responsabilidad de identificar y manipular los objetos. No es necesario algoritmos complejos, ni tampoco sensores extra.

Otro aspecto a destacar es que el diseño de estos dispositivos es distinto que otros del campo de la robótica. Por ejemplo, la apariencia del robot debería ser agradable y no parecer “una máquina” como un robot convencional. Además, se deberán tomar precauciones de seguridad porque estos robots trabajan cerca del entorno del usuario final. Por otro lado la interfaz tendrá que estar adaptada y será distinta de las que se usan en el área industrial.



2.1.1. Tipos de robots de asistencia

Diferentes objetivos y enfoques han dado como resultado hoy en día una gran variedad de robots de asistencia. Para mayor claridad podemos dividir en tres categorías los dispositivos robóticos de asistencia disponibles:

1. Robots fijos, para entornos bien estructurados.
2. Robots móviles de asistencia, que se mueven en un entorno cerrado y tienen un brazo manipulador.
3. Brazos articulados montados en una silla de ruedas que dan asistencia en cualquier lugar.

Robots fijos

Las primeras aplicaciones en los robots de asistencia se enfocaban en el uso de robots industriales y su modificación para usos de asistencia. Un factor que limita el uso de robots industriales es la diferencia de necesidades. Los brazos articulados industriales están diseñados para trabajar a altas velocidades y con gran precisión en un entorno donde no hay seres humanos. Para un área de trabajo donde hay personas los robots de asistencia deben estar mecánicamente limitados a bajas velocidades y aceleraciones. Además, la rigidez y precisión de los robots industriales no es necesaria para los robots de asistencia.

Un ejemplo de robots fijos es el proyecto de “Ayuda robótica”, un intento para crear un sistema para personas con tetraplejía mediante el uso de un robot industrial. En este proyecto se integró el brazo PUMA 250. Dificultades con el reconocimiento de voz y deficiencias computacionales limitaron el éxito del programa. La potencia de cálculo de la época no permitía la cinemática inversa del brazo en tiempo real, lo que limitaba al brazo a acciones preprogramadas. El movimiento de articulaciones individuales era posible, pero movimientos multi-articulares en tiempo real eran imposibles.¹

¹ Design, construction and testing of a wheelchairmounted robotic arm. Krvin D.Edwards

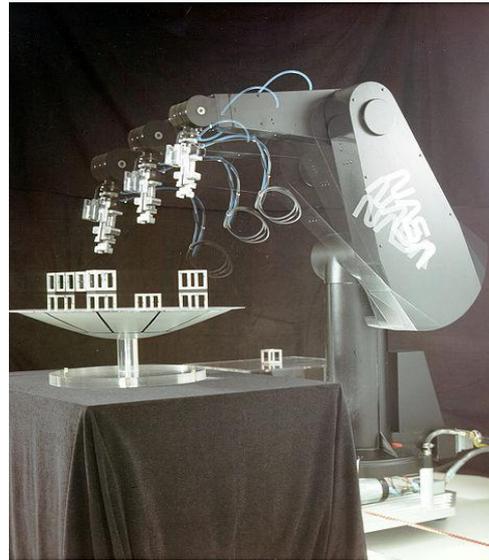


Fig. 4 PUMA 250

A medida que la disponibilidad de robots específicos fue aumentando y la potencia de cálculo de los computadores creció, los brazos con controladores pudieron ser montados en plataformas móviles. Al principio, eran simples bases rodantes, pero más tarde aumentaron en complejidad y grados de libertad para incluir potentes robots móviles.

Handy es un brazo articulado montado sobre un soporte tipo mesa que fue diseñado para ayudar en actividades muy específicas de la vida real. Handy-1



Fig. 5 Handy 1



fue desarrollado en 1987 y fue un robot muy exitoso debido a su bajo costo y sencillez de uso capaz de ayudar a personas gravemente discapacitadas en tareas cotidianas como comer, beber, lavarse, limpiar los dientes y afeitarse. Handy-1 (Fig. 5) está basado en un robot industrial de 5 grados de libertad ligeramente modificado. Consiste en un brazo robot muy sencillo que dispone de una cuchara como elemento terminal o mano, el soporte para colocar la bandeja de comida y el soporte del vaso, que también se activa para acercarse a la boca del usuario. El control se basa en accionar al robot para que acerque en el momento deseado la cuchara al usuario después de llenarla en la bandeja, a una posición próxima a la boca y previamente programada. Los intervalos de actuación son controlables por el usuario mediante un interfaz que puede ser específico según su movilidad residual. Otra orden de accionamiento desplaza el vaso hacia la boca y lo inclina para que el usuario pueda beber. El sistema Handy1 permite además intercambiar módulos para el lavado, afeitado y limpieza de dientes ofreciendo una mayor variedad de tareas a realizar por el usuario.²

Este dispositivo de asistencia no elimina la necesidad de un asistente personal, pero permite que las personas tengan un mayor nivel de autosuficiencia.

Otro ejemplo es el sistema RAID, una estación de trabajo que cuenta con el soporte de un robot para acercar objetos al usuario o realizar operaciones como archivar documentos, pasar hojas de un libro, cargar un disquete en el computador, etc. El robot se desplaza sobre unas guías verticales y horizontales para alcanzar todos los puntos de interés. El sistema es controlado por el usuario a través de un *joystick*, que además es utilizado para operar con el computador. A través de un menú en pantalla, el usuario puede controlar un gran número de elementos y efectuar distintas operaciones en la estación de trabajo.

El sistema RAID se beneficia en gran medida de la estructura formal de la estación de trabajo. Esta organización permite que el robot se mueva

² An Overview of the Development of Handy 1, a Rehabilitation Robot to Assist the Severely Disabled. MIKE TOPPING



repetidamente y adquiera los artículos necesarios por el usuario utilizando las funciones preprogramadas y las rutinas.

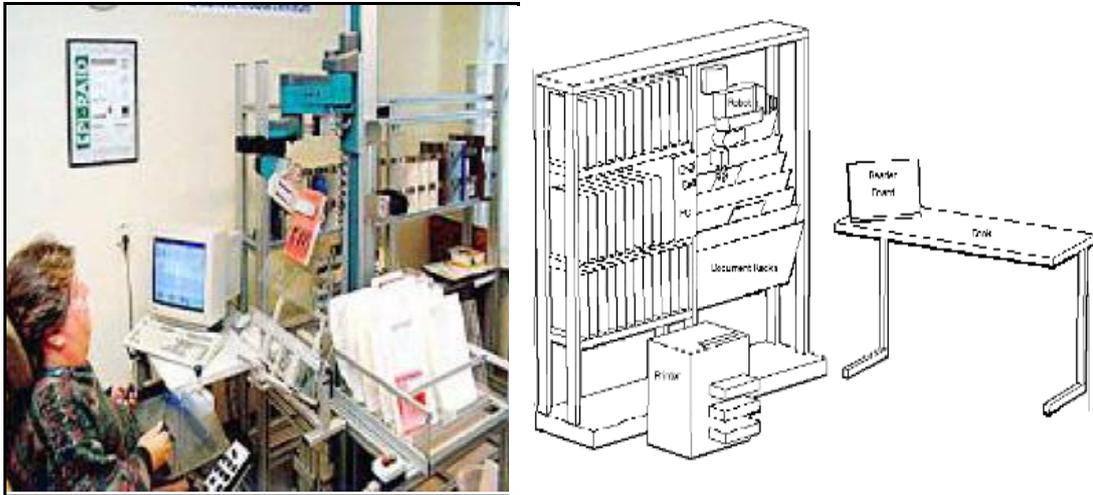


Fig. 6 Raid

Un sistema similar es el ProVar, que está basado en el robot PUMA 260, y ha sido diseñado para funcionar en un entorno profesional. El manipulador ProVar es la siguiente generación del sistema DeVar y se expande en la investigación al reducir los costos operativos y aumentar la utilidad general.

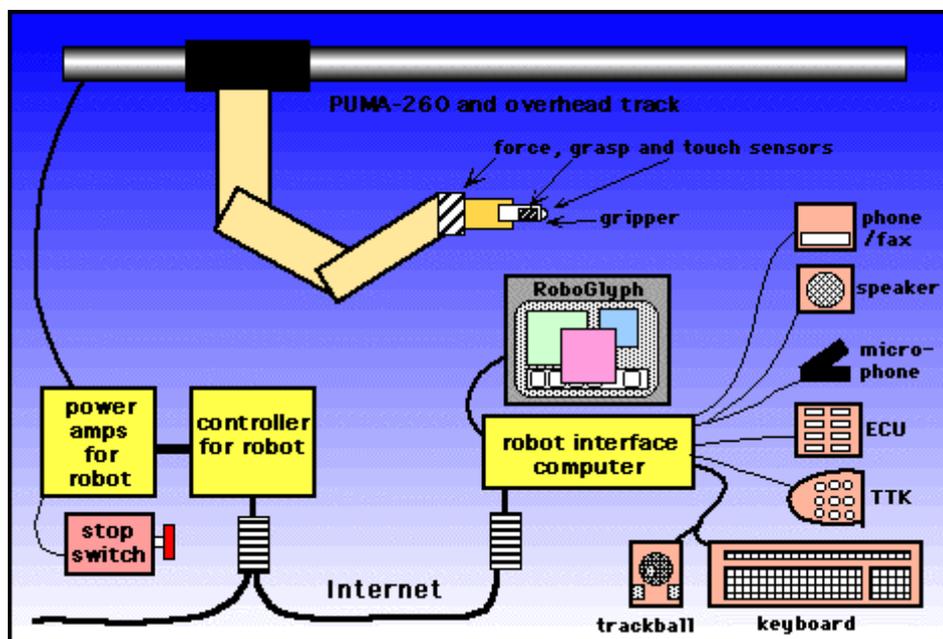


Fig. 7 Sistema ProVar

El sistema ProVar utiliza un entorno virtual basado en web para modelar la funcionalidad del robot. De esta forma el operador puede examinar los



posibles movimientos del brazo cuando le ha sido asignada una tarea y si la simulación es correcta la tarea puede ser realizada. Los principales objetivos del ProVar son una mayor funcionalidad, control más sencillo y una mayor fiabilidad del sistema en comparación con la anterior generación de robots de asistencia.³

Robots móviles

Uno de los robots móviles más conocidos en este ámbito es el robot “Mobile Vocational Assistant Robot” (MoVAR), utiliza una plataforma móvil omnidireccional y monta un brazo robótico PUMA-250, cámaras remotas, sensores de fuerza y sensores de proximidad.

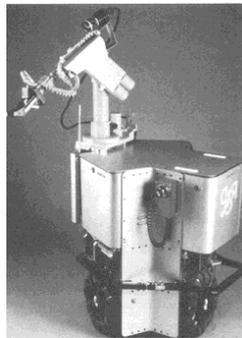


Fig. 8 MoVAR

MoVAID es una versión avanzada del sistema MoVAR que está diseñado específicamente para el uso doméstico. MoVAID incrementa la eficacia del modelo anterior. MoVAID incorpora una serie de dispositivos de detección montados tanto en la base como en el brazo.⁴



Fig. 9 MoVAID

³ MANUS --a wheelchair-mounted rehabilitation robot. B J F Driessen¹, H G Evers² and J A v Woerden¹.

⁴ Rehabilitation and Health Care Robotics.H.F. Machiel Van der Loos, David J. Reinkensmeyer



Robots montados en sillas de ruedas

Por último tenemos los brazos articulados que pueden ir montados en sillas de ruedas. Estos robots combinan la idea de los robots fijos y los móviles. En el pasado los robots eran demasiado pesados para ir montados en sillas de ruedas, pero actualmente el peso no es un impedimento.

De este tipo de robots los más conocidos o comercialmente disponibles son el iArm, distribuido por Exact Dynamics, y el Raptor distribuido por Applied Resources.

El Raptor tiene cuatro grados de libertad más una pinza plana y puede ser montado en una silla de ruedas. El usuario controla directamente el brazo ya sea con un joystick o con un controlador de 10 botones. Este brazo articulado no tiene encoders, por lo que minimiza los costes generales y lo hace más asequible.⁵



Fig. 10 Raptor

Otro robot que podríamos introducir en esta categoría es el ASIBOT. El ASIBOT es un robot manipulador con 5 grados de libertad, de unos 10kg de peso, 1,3m de alcance y 2kg de carga. Lo que le diferencia de otros robots es el sistema de anclajes del que va equipado. En cada punta está equipado con conectores especiales cónicos. Mediante estos conectores, una de las

⁵ Robotics in Rehabilitation Engineering. Michael Hillman



extremidades puede anclarse a un punto de anclaje o *'docking station'* que pueden estar situados estratégicamente en el entorno de manera que el robot pueda desplazarse de uno a otro. Esto unido a que puede ser acoplado a una silla de ruedas le otorga una gran movilidad.⁶



Fig. 11 ASIBOT

2.1.2. iArm

Exact Dynamics es la compañía que produce el brazo articulado iArm, diseñado para ofrecer cierta independencia a personas con discapacidades. El iArm es un robot que puede ir montado en una silla de ruedas y ha sido diseñado para poder manipular los objetos dentro de su área de trabajo controlado por un dispositivo de entrada. Su estructura con la base cilíndrica y telescópica le permite alcanzar objetos, además de realizar diferentes tipos de manipulación en su entorno. Este robot tiene definida una serie de requisitos:

1. Tiene un tamaño compacto, para que cuando esté montado a la silla de ruedas pueda pasar por los sitios, por ejemplo, entre puertas.
2. Es poco pesado, para minimizar la carga en la silla de ruedas.
3. Puede ser fácil de controlar con dispositivos de entrada como *joysticks*, pantallas táctiles o teclados.
4. Está mecánicamente integrado en la silla de ruedas.

⁶ A Portable Light-weight Climbing Robot for Personal Assistance Applications. A. Gimenez, A. Jardon, R. Correal, R. Cabas, and C. Balaguer



5. El control del usuario final está integrado con el control de la silla de ruedas.
6. El consumo de energía es bajo.



Fig. 12 IArm

El resultado del diseño es un brazo articulado con 6 grados de libertad, por lo que con la pinza que dispone puede alcanzar cualquier posición en el espacio y adoptar la orientación necesaria. Todas las articulaciones están impulsadas por motores de 24 V, que están localizados en la base. La ubicación de los motores permite que el brazo tenga un tamaño pequeño y baja masa de modo que los pares en cada articulación y el impulso son mínimos en caso de colisión. Como los motores no se encuentran en el propio brazo, el accionamiento electrónico puede ser integrado y los cables no limitan el número de rotaciones en la misma dirección. Cada motor tiene su propio engranaje para proporcionar el par necesario.

El brazo puede levantar 15N (1.5 kg). La transmisión de la energía se logra por medio de correas dentadas. En el hombro, se utiliza un sistema de ejes huecos concéntricos, mientras que en el codo hay una unidad de engranaje cónico. Los movimientos de la muñeca se realizan a través de un sistema mecánico diferencial.



El robot está equipado con dos dedos en la pinza que se abre y cierra mediante un resorte. Cuando el usuario cierra la pinza, se aplica una tensión en el resorte. Si la pinza toca un objeto, el usuario puede seguir cerrando la pinza incrementando su fuerza. La pinza tiene un sistema que garantiza que los objetos son manipulados con una compresión de tres puntos. Otra ventaja que tiene esta pinza es que si hay un corte de energía la pinza se mantiene cerrada.

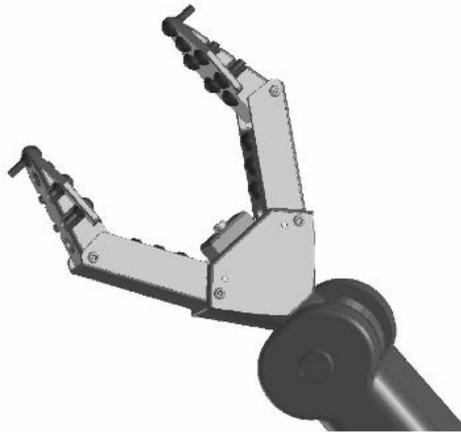


Fig. 13 Pinza del iARM

Para controlar la posición del manipulador, se utiliza un *encoder* incremental. Los *encoders* están integrados en el motor y envían 400 pulsaciones/rotación. De esta manera se pueden obtener altas resoluciones en las medidas de posición del extremo del eje. Además, se utilizan *encoders* absolutos durante el arranque para inicializar los *encoders* incrementales, y para la detección del deslizamiento.

2.2. Interfaces para personas con discapacidades

La accesibilidad es un reto para personas con discapacidad. Las personas con discapacidad pueden tener una amplia gama de puntos fuertes y débiles. Estas diferencias pueden darse dependiendo de la capacidad cognitiva, características sensoriales, la destreza motora, habilidades de comportamiento y habilidades sociales.

Cada persona es única y a menudo requiere su propia solución personalizada. Es por ello que no se puede generalizar sobre la población.



Cuando se habla de Interacción Persona-Computador en el área de la Tecnología Asistencial se tienen que tener en cuenta las distintas discapacidades. Así pues, las líneas de trabajo para realizar la interfaz que se proponen están inspiradas en la investigación de áreas como la usabilidad, la interacción persona computador, e interfaces de usuario adaptadas.

1. La interfaz debe ser fácil de usar.
2. Los procesos de las interfaces deben ser lo más cortos posibles para minimizar la sobrecarga cognitiva.
3. La interfaz debe ajustarse a distintas habilidades o capacidades del usuario.
4. La interfaz debe aprovechar las habilidades sensoriales del usuario para mejorar el control.
5. La interfaz debe adaptarse a múltiples dispositivos de acceso.

2.2.1. Interfaces para el iArm

El iArm es un brazo articulado que puede ir montado en una silla de ruedas que funciona en entornos no estructurados. Es controlado a través de los menús de un teclado, un *joystick* o por un interruptor. Estos controles no son intuitivos o naturales, ya que la reacción entre la acción del usuario y el movimiento del robot no es directa. Cada movimiento requiere un número de acciones no intuitivas que necesitan entrenamiento. Además, puede que estos dispositivos de entrada no sean acordes a las capacidades físicas del usuario.



Fig. 14 Dispositivo para controlar el iArm



Muchas instituciones están investigando interfaces más intuitivas para el iArm. En el instituto de tecnología de New Jersey han propuesto tres alternativas: una caja sensorial de infrarrojos, un lápiz con articulaciones imitando el brazo del robot y un ratón de ordenador. En TNO Science & Industry la universidad tecnológica de Delft, el iArm ha sido equipado con cámaras, sensores de fuerza y sensores infrarrojos de distancia. Es dirigido por un *joystick* y un interruptor, compartiendo la autonomía entre el brazo y el usuario.

Otra solución es la que ofrece INRIA (Institut National de Recherche en Informatique et en Automatique), que explora una propuesta basada en la visión por computador. En esta propuesta el brazo del robot está equipado con dos cámaras, una situada en el brazo que ofrece una visión fija del espacio de trabajo y otra que ofrece una visión más detallada. Cuando el usuario hace clic en el objeto deseado el brazo se mueve hacia el objeto utilizando un esquema de control visual.

Una propuesta parecida es la que siguen en la universidad de Massachussets Lowell basada en una interfaz visual que dispone de un solo interruptor que utiliza un video con rectángulo superpuesto. Cuando el recuadro contiene el objeto deseado el usuario hace clic en el interruptor para mover el brazo a la posición XY deseada.⁷



Fig. 15 Selección de objetos usando la interfaz propuesta por la universidad de Massachussets Lowell

⁷ Development and Evaluation of a Flexible Interface for a Wheelchair Mounted Robotic Arm. Katherine Tsui And Holly Yanco





3. Hardware y software usado en el proyecto final de carrera

En este capítulo se van explicar las herramientas utilizadas, tanto a nivel de *hardware* como de *software*. Se describirán por un lado los componentes de hardware como son el robot, joystick, PC y conexiones. Por otro lado se tratará de explicar las distintas librerías y herramientas utilizadas para el desarrollo del proyecto final de carrera.

3.1. Hardware

Las aplicaciones que se han creado en este proyecto permiten al usuario manejar el robot iArm con un PC compatible usando un "CAN-USB interface".

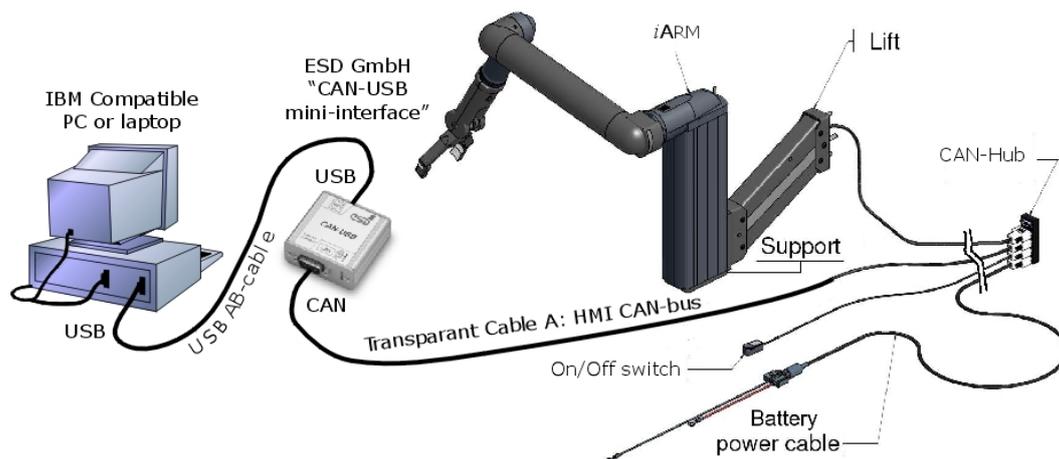


Fig. 16 Esquema de comunicación iArm

En la Fig.16 se puede ver el esquema para el uso de las aplicaciones. El PC se comunica con el iArm a través de un CAN-bus (Controller Area Network), mediante un USB-CAN interface de ESD ELECTRONICS GMB. En este esquema falta poner el *joystick* conectado al PC, que se usa en una de las interfaces.

3.1.1. El robot iArm

El robot iArm es un brazo articulado fabricado por la compañía Exact Dynamics. Este robot tiene el objetivo de ofrecer cierta independencia a



personas con discapacidades en sus extremidades. En el capítulo (2.1.2) se ha explicado con más detalle las cualidades de este producto. A continuación podemos ver una lista con sus características técnicas.

- Peso máximo que puede levantar: 1,5 kg.
- Área de trabajo del brazo iArm: Esférico, un radio de 80 cm aproximadamente.
- Pinza: Dos dedos paralelos de cierre.
- Fuerza de sujeción máxima: Aproximadamente 20N.
- Apertura máxima de la pinza: Aproximadamente 9,9 cm.
- Velocidad angular máxima de las articulaciones:
 - Ejes 1, 2, 3: 0,35rad/sec.
 - Ejes 4, 5, 6: 0,70 rad/sec.
- Velocidad cartesiana máxima: 100mm/sec.
- Fuente de alimentación: 21 a 34 V CC. 1A de media, máximo 3A
- Área de trabajo del elevador: 20cm verticalmente.
- Peso del elevador: 2,9kg.
- Pantalla de visualización: Matriz x7 de LED con timbre.

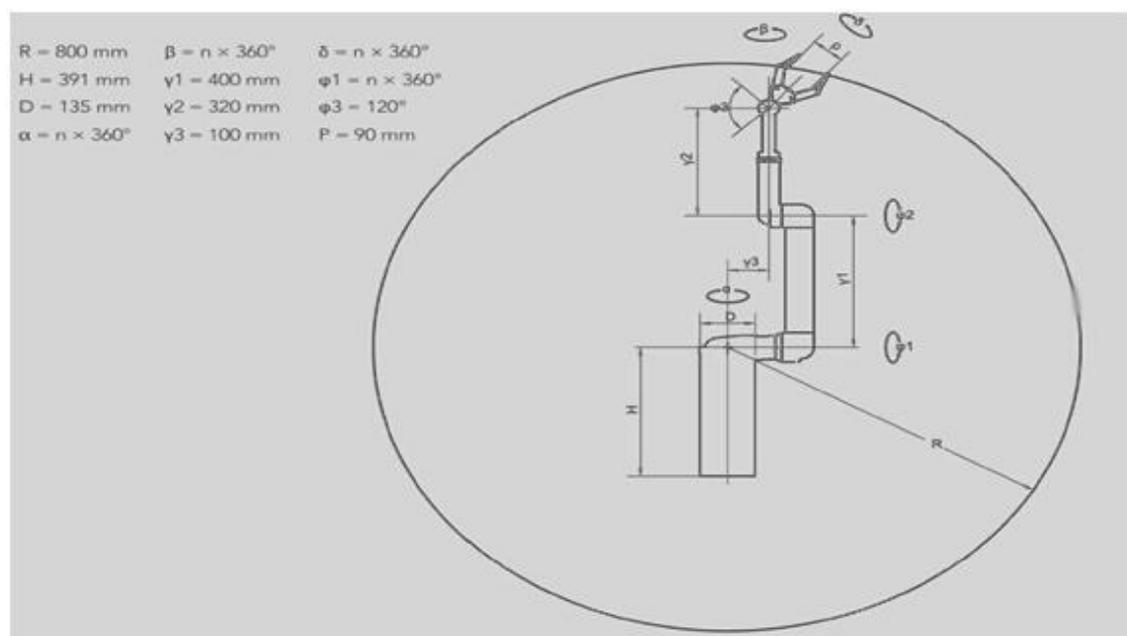


Fig. 17 Área de trabajo del iArm



3.1.2. USB-CAN interface

Como se ha explicado anteriormente la conexión del brazo articulado con el PC se hace mediante un CAN-bus. CAN (acrónimo del inglés Controller Area Network) es un protocolo de comunicaciones desarrollado por la firma alemana Robert Bosch GmbH, basado en una topología bus para la transmisión de mensajes en entornos distribuidos. Es un protocolo de comunicaciones normalizado, con lo que se simplifica y economiza la tarea de comunicar subsistemas de diferentes fabricantes sobre una red común o *bus*. El procesador anfitrión (*host*) delega la carga de comunicaciones a un periférico inteligente, por lo tanto el procesador anfitrión dispone de mayor tiempo para ejecutar sus propias tareas. Otra ventaja es que, al ser una red multiplexada, reduce considerablemente el cableado y elimina las conexiones punto a punto.

En este proyecto se utiliza un USB-CAN-Interface mini de ESD Electronics.

Información técnica:

- Nombre: USB-CAN-Interface para USB 1.1
- Tipo: Adaptador USB-CAN, Interfaz CAN con aislamiento eléctrico, 1MBit/s, ISO11898.
- Dimensiones de la caja: 55mm x 55mm x 25mm.
- MicroControlador: MB90F543 con controlador CAN integrado.
- Controlador USB: PDI USBD12, 12Mbit/s.
- Fuente de alimentación: via USB.



Fig. 18 USD-CAN interface



3.1.3. Joystick

Una de las interfaces desarrolladas para este proyecto final de carrera hace uso de un *joystick*. Se puede hacer uso de cualquier tipo de *joystick*, siempre que el sistema operativo que utilicemos (Linux o Windows) sea capaz de reconocerlo. Es decir, tendremos que tener instalados correctamente los *drivers* del *joystick* que vayamos a utilizar.

Respecto al modelo, no hay limitaciones. Se puede usar un dispositivo con cualquier número de botones y ejes. La aplicación es la que se adaptará al tipo de *joystick* que esté conectado.

A pesar de no haber una limitación concreta del tipo de *joystick* a usar, para tener un control total sobre el brazo articulado necesitaremos un determinado número de botones o ejes. Además si el *joystick* dispone de *feedback* (retroalimentación), proporcionará otra funcionalidad añadida a la interfaz.

Como se trata de una interfaz que tiene que adaptarse al usuario final, la complejidad del *joystick* escogido deberá ser en función de las habilidades que disponga el usuario.

La interfaz se ha hecho con la idea de poder utilizar cualquier *joystick* compatible del mercado. Para el desarrollo se ha utilizado un *joystick* de la marca Logitech, en concreto el "Force 3D PRO". Con sus 12 botones, 3 ejes y el botón direccional podemos tener un control total de todas las funcionalidades que nos ofrece el brazo articulado. Además, tiene '*forcefeedback*', que aporta una funcionalidad añadida.



Fig. 19 Joystick. Force 3D PRO



3.2. Software

En este apartado se va a describir con más detalle el tipo de lenguaje de programación y las bibliotecas que se han utilizado para el desarrollo de las dos interfaces.

3.2.1. C++ en el proyecto

C++ es un lenguaje procedural (orientado a algoritmos) y orientado a objetos. Como lenguaje procedural se asemeja al C y es compatible con él. Pero como lenguaje orientado a objetos se basa en una filosofía completamente diferente, que exige del programador un completo cambio de mentalidad. Las características propias de la Programación Orientada a Objetos (Object Oriented Programming, u OOP) de C++ son modificaciones mayores que sí que cambian radicalmente su naturaleza.

En cuanto a la razón principal por la que se utiliza C++ como lenguaje de programación en este proyecto es porque las librerías utilizadas para realizar las dos interfaces son compatibles con este lenguaje de programación.

3.2.2. Compilador MinGW

Existen diversos compiladores para C++, uno de los compiladores libres es el de GNU, el compilador G++ (parte del proyecto GCC, que engloba varios compiladores para distintos lenguajes). Otros compiladores comunes son Intel C++ Compiler, Xcode, Borland C++, CodeWarrior C++, g++ de Cygwin, g++ de MinGW, Visual C++, Carbide.c++, entre otros.

Para nuestro proyecto hemos utilizado el compilador g++ de MinGW adecuado al sistema de operativo de Windows.

MinGW (Minimalist GNU for Windows), anteriormente conocido como MinGW32, es una implementación de los compiladores GCC para la plataforma Win32. Además MinGW incluye un conjunto de la API (Application Programming Interface) de Win32, permitiendo un desarrollo de aplicaciones nativas para esa plataforma, pudiendo generar ejecutables y bibliotecas usando la API de Windows.



3.2.3. Qt Creator como entorno de desarrollo

En un principio se pensó usar NetBeans para C++ como entorno de desarrollo. NetBeans es una potente herramienta para el desarrollo de aplicaciones que además posee un número importante de módulos que pueden ser extendidos. Entre ellos se encuentra un módulo para hacer uso de la librería QT que se utiliza en el proyecto para crear las interfaces gráficas y que se describirá más tarde.

Qt Creator es un IDE creado por Trolltech para el desarrollo de aplicaciones con las librerías Qt. Los sistemas operativos que soporta en forma oficial son:

- GNU/Linux 2.6.x, para versiones de 32 y 64 bits con Qt 4.x instalado. Además hay una versión para Linux con gcc 3.3.
- Mac OS X 10.4 o superior, requiriendo Qt 4.x
- Windows XP y superiores, requiriendo el compilador MinGW y Qt 4.4.3 para MinGW.

La elección de Qt Creator y no NetBeans como entorno de desarrollo se debió a que la interfaz de Qt Creator para hacer uso de la librería Qt era más fácil de usar que la que la de NetBeans.

3.2.4. Qt, la librería

Como se ha comentado en el apartado anterior, la razón por la que se optó por usar Qt Creator es el uso de las librerías Qt. Qt es una librería multiplataforma ampliamente usada tanto para el desarrollo de aplicaciones con una interfaz gráfica de usuario, como para el desarrollo de programas sin interfaz gráfica como herramientas para la línea de comandos y consolas para servidores.

Qt es utilizada en KDE, un entorno de escritorio para sistemas tales como GNU/Linux o FreeBSD, entre otros. Qt utiliza el lenguaje de programación C++ de forma nativa, aunque adicionalmente puede ser utilizado en otros varios lenguajes de programación a través de *bindings*. También es usada en sistemas informáticos empotrados para automoción, aeronavegación y aparatos domésticos tales como frigoríficos.



Esta biblioteca funciona en todas las principales plataformas, y tiene un amplio apoyo. El API de la biblioteca cuenta con métodos para acceder a bases de datos mediante SQL, así como uso de XML, gestión de hilos, soporte de red, una API multiplataforma unificada para la manipulación de archivos y una multitud de otros métodos para el manejo de ficheros, además de estructuras de datos tradicionales.

Distribuida bajo los términos de GNU Lesser General Public License (y otras), Qt es software libre y de código abierto. Es producido por la división de software Qt de Nokia, que entró en vigor después de la adquisición por parte de Nokia de la empresa noruega Trolltech, el productor original de Qt.

Algunos ejemplos de aplicaciones populares que utilizan Qt son: AdobePhtoshopAlbum, Google Earth, Skype, VIC Media Player, VirtualBox...

En este proyecto el uso de la librería Qt está muy presente en las dos interfaces creadas. En las que se usa para el desarrollo de las interfaces gráficas de usuario.

3.2.5. API iARM

El iARM incluye una interfaz de programación de aplicaciones o API (Application Programming Interface). Esta API tiene definidas unas funciones para controlar el brazo articulado.

Las interfaces creadas en este proyecto hacen uso de las funciones definidas en la API del iARM para controlar el robot. Lo único que hay que hacer para poder usarlas es abrir la conexión con el iARM. Se puede encontrar más información acerca de las funciones que están definidas en la documentación del iARM Transparent Mode.

3.2.6. SDL

Por último tenemos las librerías SDL que han sido utilizadas para desarrollar la interfaz que hace uso del *joystick*. Simple DirectMedia Layer (SDL) es un conjunto de bibliotecas desarrolladas en el lenguaje de programación C que proporcionan funciones básicas para realizar operaciones de dibujo en dos dimensiones, gestión de efectos de sonido y música, además



de carga y gestión de imágenes. Fueron desarrolladas inicialmente por Sam Lantinga, un desarrollador de videojuegos para la plataforma GNU/Linux.

Pese a estar programado en C, tiene *binds* a otros lenguajes de programación tales como C++, Ada, C#, BASIC, Erlang, Lua, Java, Python, etc. También proporciona herramientas para el desarrollo de videojuegos y aplicaciones multimedia. Una de sus grandes virtudes es el tratarse de una biblioteca multiplataforma, siendo compatible oficialmente con los sistemas Microsoft Windows, GNU/Linux, Mac OS y QNX, además de otras arquitecturas y sistemas como Sega Dreamcast, GP32, GP2X, etc.

La biblioteca se distribuye bajo la licencia LGPL, que es la que ha permitido el gran avance y evolución de SDL.

SDL es una gran herramienta para el desarrollo de juegos, pero lo que le hace interesante para este proyecto es la posibilidad de acceder a los recursos del sistema. En concreto, nos da la facilidad para acceder al *joystick* y poder utilizar sus funciones para el desarrollo de la interfaz que hace uso del *joystick*.

En principio se utilizó la versión SDL 1.2, pero con esta no podíamos acceder a las funcionalidades del *forcefeedback* que nos ofrece el *joystick*, por lo que para este proyecto se utiliza la versión SDL 1.3 a pesar de no ser estable, pero que ofrece la posibilidad de utilizar las funciones que hacen uso del *forcefeedback*.



4. Desarrollo de un sistema de entrenamiento

En este apartado se va explicar cómo se ha desarrollado el sistema de entrenamiento para el brazo articulado portátil. En concreto explica cómo funcionan las dos aplicaciones para las dos interfaces de las que se compone el sistema de entrenamiento.

4.1. Introducción

El sistema de entrenamiento que se propone para este proyecto consiste en dos interfaces para el uso del brazo articulado iARM. Pero no se trata solo de dos interfaces para controlar el robot. Es un sistema en el que el usuario final es el que decide la interfaz y configuración que se adapta mejor a sus habilidades.

El proyecto está enfocado en los usuarios finales que van hacer uso de las interfaces. Está dirigido a usuarios que tienen algún tipo de discapacidad motora, pero sin determinar qué tipo de movimiento controlado mantienen. Por lo tanto, el problema es la amplia diferencia de habilidades que puede tener el usuario final.

Teniendo en cuenta a quien va dirigido y cuáles son los problemas el objetivo de este sistema de entrenamiento es que el usuario pueda decidir qué interfaz y qué configuración se adapta mejor a sus habilidades.



4.2. Modelo de casos de uso

En este apartado se van a mostrar los casos de uso que tienen en común las aplicaciones desarrolladas. Ambas aplicaciones tendrán los mismos casos de uso para controlar el robot, la parte de configuración e inicialización será lo que varíe.

En la Fig. 20 se puede ver representado el diagrama de casos de uso de las diferentes funciones que tiene el usuario para controlar el robot.

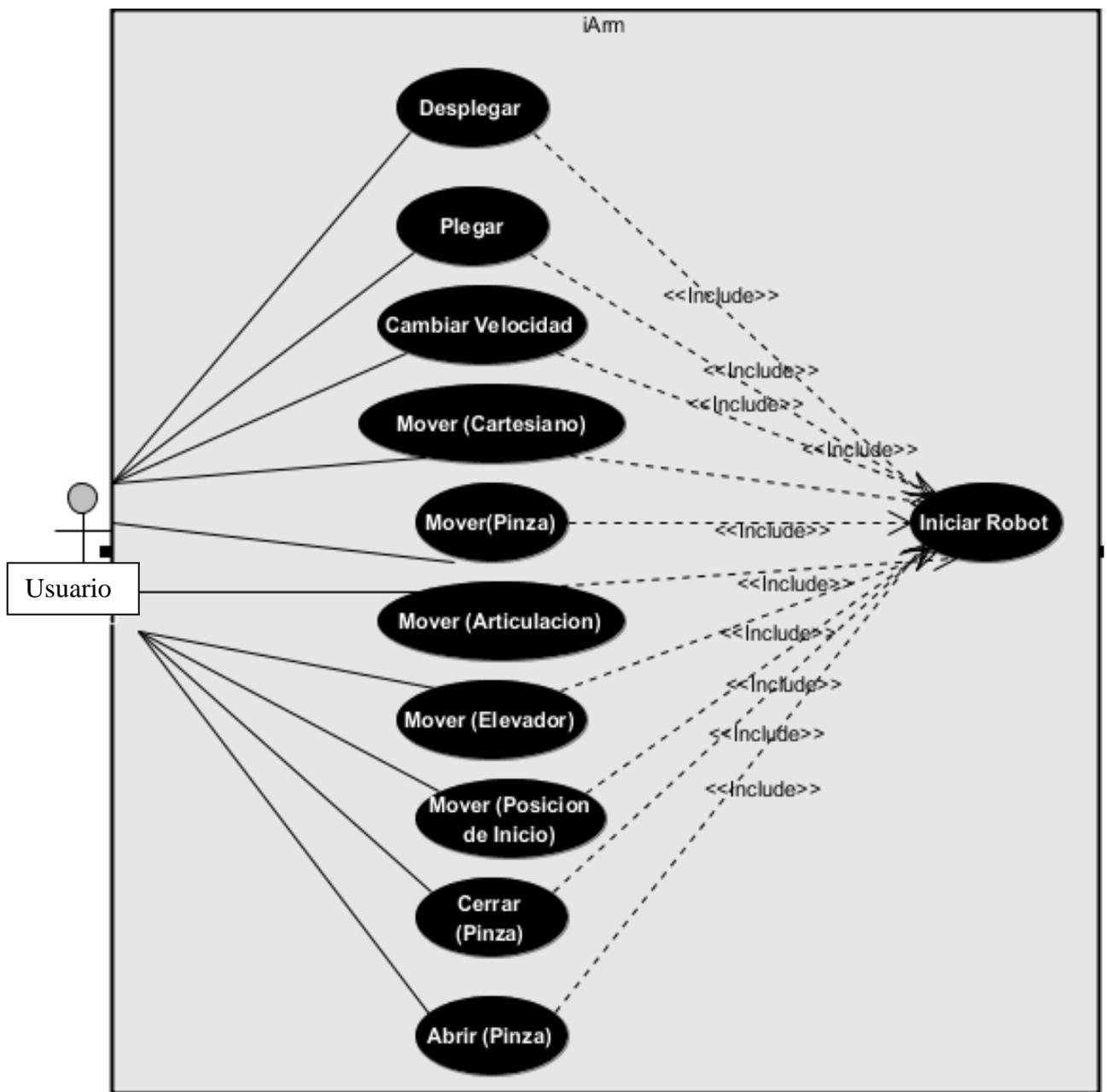


Fig. 20 Modelo de casos de uso



A continuación se explica más detalladamente los casos de uso extendidos para poder entender su funcionamiento.

4.2.1. Desplegar

<u>Descripción</u>
El usuario despliega el brazo articulado de forma automática.
<u>Flujo de eventos</u>
<ul style="list-style-type: none">• El robot está plegado o en otra posición y el usuario lo quiere desplegar para empezar a controlarlo.
<u>Requisitos</u>
<ul style="list-style-type: none">• El robot tiene que estar activo.• Tiene que estar en una posición en la que pueda desplegarse. Para no dañar el robot, hay posiciones desde las que el brazo articulado no puede llegar a la posición de despliegue.• Tiene que haber espacio libre alrededor para no dañar el brazo articulado.
<u>Imágenes</u>

Tabla 2 Caso de uso: Desplegar



4.2.2. Plegar

<u>Descripción</u>
El usuario pliega el brazo articulado de forma automática.
<u>Flujo de eventos</u>
<ul style="list-style-type: none">• El robot está desplegado o en otra posición y el usuario lo quiere plegar para recogerlo.
<u>Requisitos</u>
<ul style="list-style-type: none">• El robot tiene que estar activo.• Tiene que estar en una posición en la que pueda plegarse. Para no dañar el robot, hay posiciones desde la que el brazo articulado no puede llegar a la posición de pliegue.
<u>Imágenes</u>


Tabla 3 Caso de uso: Plegar

4.2.3. Cambiar Velocidad

<u>Descripción</u>
El usuario puede cambiar la velocidad con la que controlar los movimientos del brazo articulado.
<u>Flujo de eventos</u>
<ul style="list-style-type: none">• En cualquier momento una vez iniciado el robot puede cambiar la velocidad.
<u>Requisitos</u>
<ul style="list-style-type: none">• El robot tiene que estar activo.



- No a todos los movimientos se les puede cambiar la velocidad. En general serán los movimientos manuales como los movimientos cartesianos, articulares, de orientación de la pinza, de abrir y cerrar pinza a los que se les podrá cambiar la velocidad.

Tabla 4 Caso de uso: Cambiar Velocidad

4.2.4. Mover (Cartesiano)

<u>Descripción</u>
Mueve el punto de centro de herramienta (TCP) de la ubicación en la que está con respecto a la base en múltiples direcciones (sobre las coordenadas x, y, z).
<u>Flujo de eventos</u>
<ul style="list-style-type: none">• El usuario mueve el punto de centro de herramienta (TCP) .El TCP está situado en la pinza. Puede moverlo simultáneamente sobre múltiples direcciones a la velocidad establecida.• Si se intenta ejecutar un movimiento articular, del elevador, desplegar, plegar o mover a posición de inicio mientras se está haciendo un movimiento cartesiano se dejara de ejecutar el movimiento cartesiano.• El TCP no se puede mover a la velocidad especificada si se acerca o alcanza un punto singular interno, en tal caso el iARM perderá uno o más grados de libertad y tendrá que reducir su velocidad angular y linear durante el acercamiento o cuando pase por la singularidad interna para garantizar una trayectoria recta durante el movimiento.• Si el TCP alcanza una configuración singular en los límites del espacio de trabajo del robot se generará un error.
<u>Requisitos</u>
<ul style="list-style-type: none">• El robot tiene que estar activo.

Tabla 5 Caso de uso: Mover (Cartesiano)



4.2.5. Mover (Orientación de la pinza)

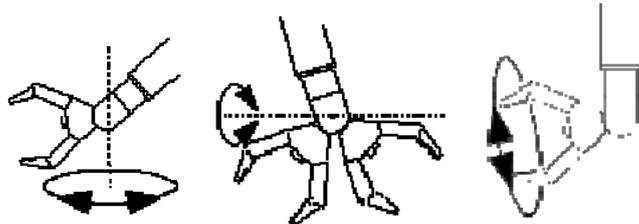
<u>Descripción</u>
Mueve el punto de centro de herramienta (TCP) de la ubicación en la que está con respecto a la base en distintas orientaciones (yaw, pitch, roll).
<u>Flujo de eventos</u>
<ul style="list-style-type: none">• El usuario mueve el TCP en la orientación deseada, yaw, pitch o roll a la velocidad indicada.• Se pueden hacer movimientos de orientación junto con movimientos cartesianos, pero no con otros movimientos.
<u>Requisitos</u>
<ul style="list-style-type: none">• El robot tiene que estar activo.
<u>Imágenes</u>


Tabla 6 Caso de uso: Mover (Orientación de pinza)

4.2.6. Mover (Articulaciones)

<u>Descripción</u>
Movimiento de cada una de las articulaciones del brazo articulado.
<u>Flujo de eventos</u>
<ul style="list-style-type: none">• El usuario puede mover cada una de las seis articulaciones de las que está compuesto el robot a la velocidad determinada.• Se pueden mover las articulaciones independientemente o simultáneamente.
<u>Requisitos</u>
<ul style="list-style-type: none">• El robot tiene que estar activo.
<u>Imágenes</u>

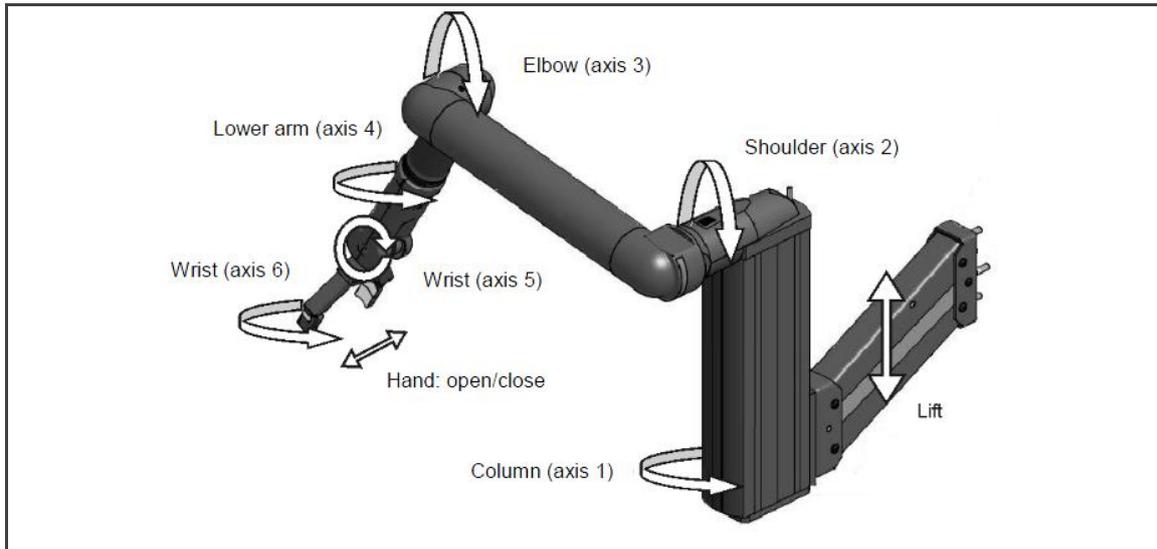


Tabla 7 Caso de uso: Mover (Articulaciones)

4.2.7. Mover (Elevador)

Descripción
Movimiento del elevador.
Flujo de eventos
<ul style="list-style-type: none">El usuario mueve el elevador en vertical hacia arriba o abajo.
Requisitos
<ul style="list-style-type: none">El robot tiene que estar activo.

Tabla 8 Caso de uso: Mover (Elevador)

4.2.8. Mover (Posición de Inicio)

Descripción
Movimiento del brazo articulado a una posición guardada en la aplicación llamada posición de inicio desde la que poder empezar a controlar el robot.
Flujo de eventos
<ul style="list-style-type: none">El usuario se encuentra una posición cualquiera y quiere llevar el robot a la posición de inicio.
Requisitos
<ul style="list-style-type: none">El robot tiene que estar activo.Si el robot esta plegado no se puede mover a la posición de inicio para evitar daños en el robot.

Tabla 9 Caso de uso: Mover(Posición de Inicio)



4.2.9. Abrir/Cerrar Pinza

<u>Descripción</u>
Abrir o cerrar la pinza del brazo articulado.
<u>Flujo de eventos</u>
<ul style="list-style-type: none">• El usuario abre y cierra la pinza a la velocidad determinada.• Puede abrir y cerrar la pinza simultáneamente con otros movimientos como los articulares o cartesianos.
<u>Requisitos</u>
<ul style="list-style-type: none">• El robot tiene que estar activo.

Tabla 10 Caso de uso: Abrir/Cerrar Pinza

4.2.10. Iniciar Robot

<u>Descripción</u>
Abrir la conexión con el brazo articulado.
<u>Flujo de eventos</u>
<ul style="list-style-type: none">• El usuario inicia el robot. Obtiene la conexión para poder empezar a controlar el iARM y mandarle las órdenes deseadas.
<u>Requisitos</u>
<ul style="list-style-type: none">• Tiene que haber una conexión entre el iARM y el PC.

Tabla 11 Caso de uso: Iniciar Robot



4.3. Sistema de entrenamiento. Parte I: Uso de una interfaz gráfica de usuario

Esta primera parte del sistema de entrenamiento es una aplicación basada en una interfaz gráfica de usuario. Para hacer la interfaz gráfica se ha utilizado la librería Qt y se han implementado todas las funcionalidades que se han descrito en el modelo de casos de uso.



Fig. 21 Sistema de entrenamiento (Parte I). Interfaz gráfica

Para esta aplicación se ha priorizado que toda la funcionalidad para controlar el brazo articulado esté a primera vista del usuario. Ayudándose de una interfaz gráfica sencilla se quiere que el usuario final pueda de una manera muy fácil e intuitiva controlar el iARM.

Para hacer uso de la interfaz gráfica se requiere un ratón y una pantalla. Con el ratón el usuario se puede mover a través de la aplicación y clicar sobre los movimientos y funcionalidades de la aplicación.

A cualquier persona que haga un uso habitual de un ordenador y este habituado a usar el ratón le resultaría sencillo controlar el iArm mediante esta interfaz gráfica, pero el usuario final de esta aplicación puede que no tenga esa



facilidad. Por ello, la aplicación también está pensada para poder ser usada con una pantalla táctil. La configuración de la pantalla y la disposición de los botones están pensadas para que se pueda controlar el brazo articulado de una manera sencilla e intuitiva. Las acciones que se hacen sobre la interfaz se hacen a través de los botones o la pestaña de inicio, exceptuando una, los movimientos cartesianos. Para mover la pinza sobre el espacio hacia adelante y a los lados se ha utilizado un recuadro que simula un 'touchpad' (alfombrilla táctil). De esta manera se ha querido conseguir que el usuario pueda mover la pinza sobre el espacio de una manera más rápida y sencilla disminuyendo el número de pasos requeridos para situar la pinza en la posición deseada.

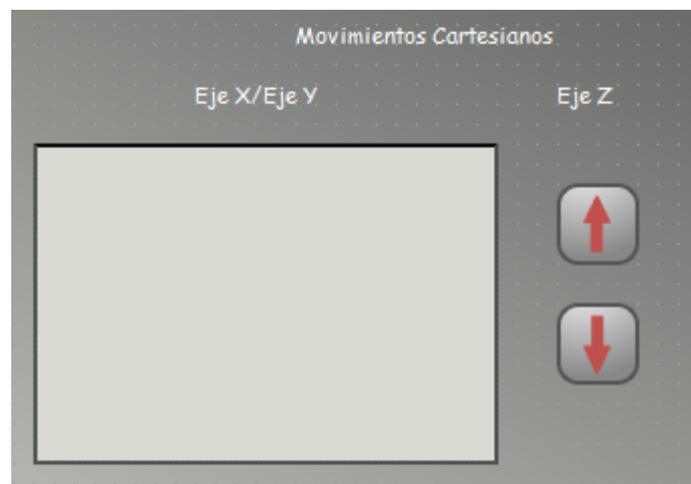


Fig. 22 Recuadro que simula un 'touchpad'

Como ya se ha comentado, toda la funcionalidad está a primera vista. Los movimientos cartesianos los podemos realizar con el 'touchpad' y los botones. A la derecha de estos tenemos tres tipos de movimientos para orientar la pinza: yaw, roll y pitch. Más a la derecha tenemos los botones acompañados de su posición articular para controlar cada una de las articulaciones individualmente. Tanto los movimientos articulares como los de orientación de la pinza tienen dos botones para realizar los movimientos en sentido positivo y negativo. Además, van acompañados con imágenes para dar una idea al usuario del movimiento que se realiza. A todos estos movimientos y a los de abrir y cerrar la pinza se les puede especificar una velocidad determinada mediante la barra de velocidad.



También se puede subir y bajar el elevador mediante los botones que se encuentran en la parte izquierda.

Mediante la pestaña de inicio que se encuentra en la parte superior izquierda de la interfaz, se desplegarán tres opciones correspondientes a tres movimientos: Plegar, Desplegar y Posición de Inicio. Estos movimientos sirven para plegar y desplegar el brazo articulado y son realizados automáticamente por el iArm.

Por último, en la parte izquierda inferior tenemos un *display* que mostrará el estado del iArm en todo momento.

La idea y diseño final de esta aplicación es que el usuario no tenga que pensar mucho ni perder tiempo sobre la interfaz para trasladar lo que quiere hacer sobre el robot y hacerlo de una manera directa. Se trata de una interfaz sobre la que no tenemos opciones personales de configuración, pero en este caso se cree que no hay muchas posibilidades de configuración. Tal vez se podrían presentar con otro diseño más intuitivo o con botones más accesibles, pero la interfaz ofrece una visión directa de las funcionalidades de las que dispone el robot, y a través de ella se pueden realizar de una manera directa y fácil las acciones que se deseen.

4.3.1. Diagrama de secuencia

La aplicación antes de iniciarse comprueba si está conectado el PC con el iARM mediante el CAN-Bus. Una vez comprobada la conexión con el robot se abre la conexión y se inicia la interfaz gráfica.

Con la interfaz abierta el usuario podrá empezar a interactuar con ella, pero para empezar a controlar el iARM primero debe encenderlo. El iARM tiene un botón para poder encender el dispositivo.

Cuando el iARM esta activado el usuario puede empezar a interactuar con la interfaz y controlar el robot.



sd Diagrama de Secuencia iARM

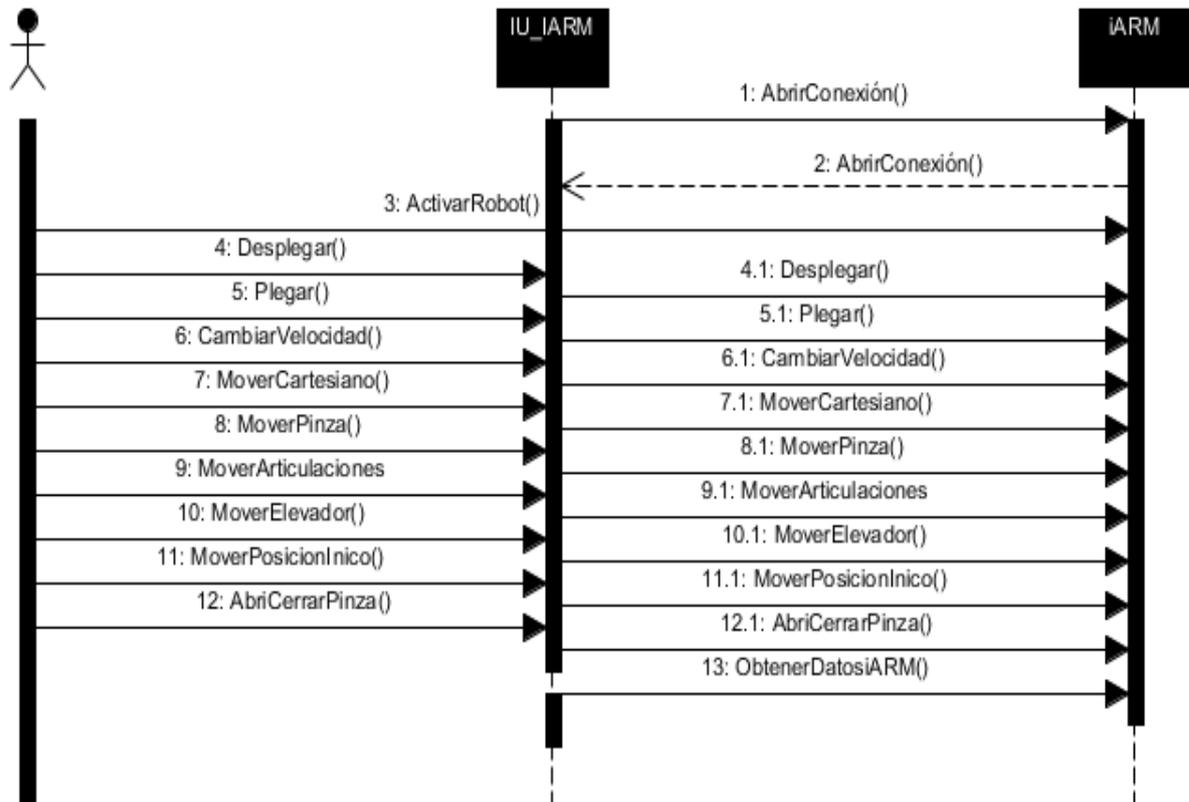


Fig. 23 Diagrama de secuencia. Sistema de entrenamiento (Parte I)

4.4. Sistema de entrenamiento. Parte II: Uso del Joystick.

La segunda parte del sistema está basada en el uso de un *joystick* para el control del brazo articulado. La aplicación tiene una parte en la que mediante una interfaz gráfica permite la configuración del *joystick*, y otra donde con el propio *joystick* se controla el brazo articulado.

A la hora de desarrollar la aplicación para el *joystick* se pensó que debía ser un sistema lo más flexible posible. La idea es no limitarse solo a un tipo de *joystick*, ni tampoco a un tipo de configuración. En el sistema de entrenamiento que se ha creado lo que se quiere es que el usuario pueda elegir la interfaz y configuración que más se adapte a sus habilidades físicas y cognitivas.

Para la selección del *joystick* que controla el robot no hay restricción, el único requisito es que el sistema operativo bajo el que se esté ejecutando la aplicación reconozca el *joystick*. El usuario podrá elegir el *joystick* que más se ajuste a sus habilidades. Debido al gran número de casos de uso que hay para



controlar el robot es recomendable usar un *joystick* que tenga bastantes botones, botones direccionales y ejes. En las pruebas hechas en el desarrollo se ha hecho con un *joystick* de 12 botones, botones direccionales y 3 ejes. Este tipo de *joystick* permite hacer una configuración en el que se pueda hacer uso de todas las funcionalidades del iARM. Pero puede haber usuarios a los que no les haga falta tener acceso a todas las funcionalidades y quieran una interfaz más sencilla e intuitiva. Por otra parte, si el *joystick* que se utiliza tiene “forcefeedback” el programa podrá hacer uso de él cuando el robot se aproxime a puntos singulares para de esta manera avisar al usuario de ellos y corregir la posición del brazo.

La interfaz (Fig. 24) permite una configuración adaptable según el *joystick* que se utilice. El programa reconoce el tipo de *joystick* que está conectado y según sus características el usuario puede elegir para cada caso de uso los botones o ejes que utiliza.

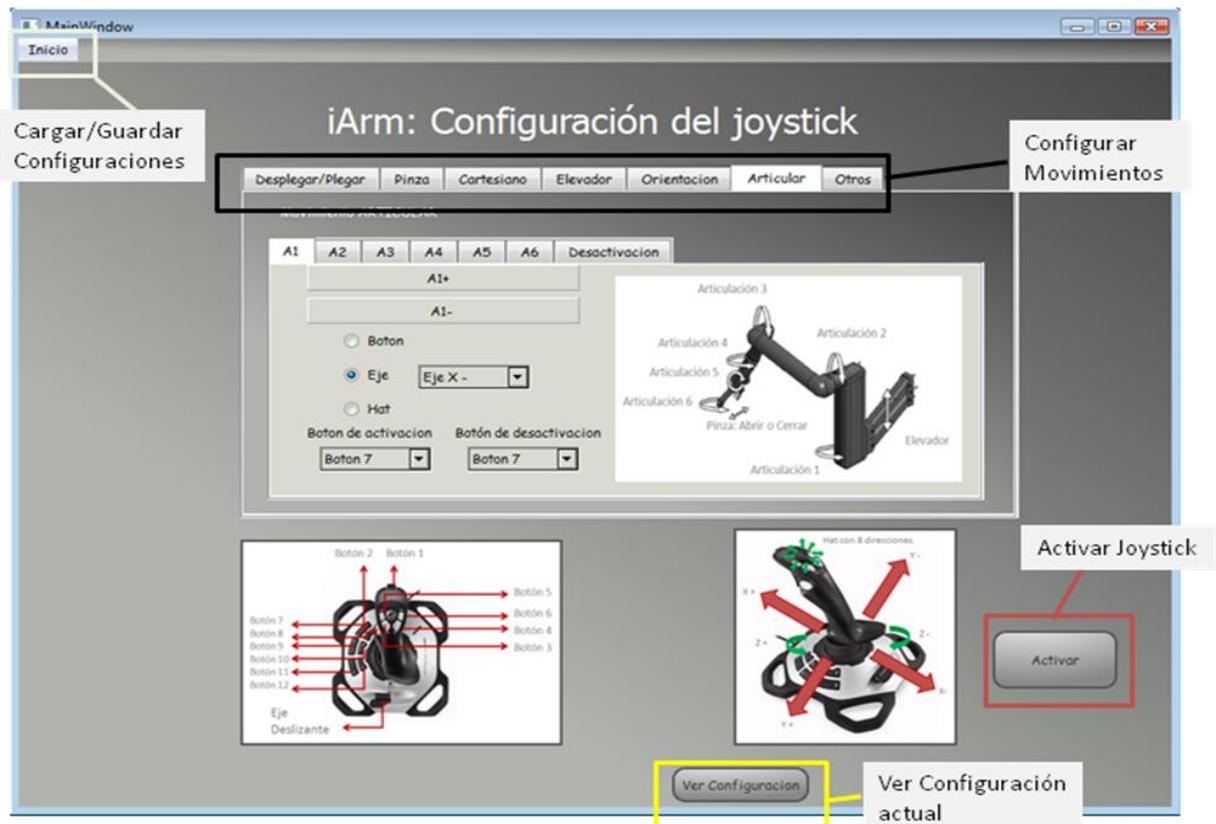


Fig. 24 Interfaz para la configuración del joystick



Para elegir que acción del *joystick* asignar a cada movimiento del iArm, la interfaz está dividida en pestañas por cada caso de uso o grupo de movimientos del iArm. En cada pestaña se tendrán los distintos movimientos que se pueden realizar con el iArm y a cada uno de ellos se le podrá asignar una acción del *joystick*. De esta manera, el usuario podrá asignar cualquier acción del *joystick* a uno o varios movimientos del brazo articulado según la configuración, ya que la interfaz permite asignar acciones del *joystick* para activar y desactivar movimientos y así asignar más de un movimiento a una acción del *joystick*.

La aplicación permite al usuario poder elegir entre una infinidad de posibles configuraciones. Por ello permite también guardar las configuraciones, para que el usuario haciendo pruebas pueda ir eligiendo y cambiando las configuraciones hasta dar con la que más se adapte a sus habilidades.

4.4.1. Diagrama de secuencia

Como se ha comentado anteriormente, en esta aplicación tiene mucha importancia la configuración que cada usuario pueda tener para manejar el *joystick* y controlar así el brazo articulado.

Primero el usuario realiza su configuración personal. La aplicación da la posibilidad de poder guardar las configuraciones y poder abrirlas. Estas configuraciones se guardarán en el sistema en archivos de texto “.txt”. Según el *joystick* que se conecte al PC el usuario tendrá la posibilidad de configurarlo según sus preferencias. Para cada caso de uso el usuario tendrá que asignarle la acción del *joystick* con la que la ejecutará.

Una vez elegida la configuración personal para manejar con el *joystick* el brazo articulado, el usuario tendrá que iniciar el iARM primero en la interfaz gráfica, y luego activando el botón de activación que tiene el iARM.

Una vez iniciado el iArm, el usuario podrá empezar a controlar el brazo articulado con el *joystick*. En pantalla, aparecerá un resumen con la configuración seleccionada.



sd Diagrama de secuencia

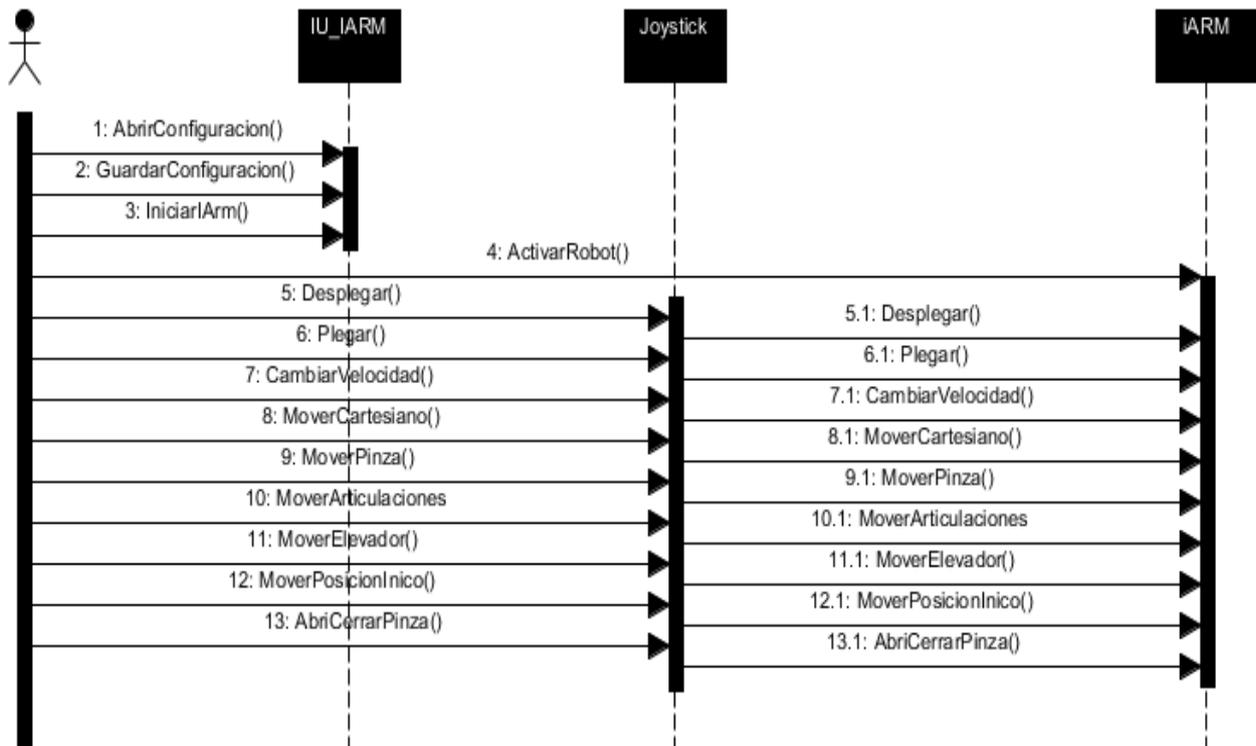


Fig. 25 Diagrama de secuencia. Sistema de entrenamiento (Parte II)





5. Revisión del documento de objetivos del proyecto

En este apartado se revisa el documento de objetivos del proyecto. En concreto se analizan los cambios que ha habido durante el desarrollo del mismo, para en proyectos futuros tener en cuenta estas consideraciones.

5.1. *Revisión de los objetivos y planificación del proyecto*

Si bien todos los objetivos no estaban definidos al inicio del proyecto, quedaron contemplados posteriormente y se han conseguido cumplir. Durante el desarrollo del proyecto ha habido pequeños cambios de requerimientos pero que no han hecho variar los objetivos finales del proyecto.

En principio la idea fue crear una interfaz avanzada, pero posteriormente se decidió crear un sistema de entrenamiento para el brazo articulado que cumpliera una serie de requisitos. Los objetivos variaron después de la replanificación, pero son los que están redactados en el apartado de documento de objetivos del proyecto. Las interfaces sí que sufrieron pequeñas modificaciones que no estaban planificadas. Fueron mejoras para completar las interfaces del sistema de entrenamiento, de las que se pueden destacar el uso del “*forcefeedback*” del *joystick*, el poder guardar las configuraciones o el uso del *touchpad* o alfombrilla en la interfaz gráfica.

El número de horas de trabajo resultantes no son las mismas que las horas estimadas en un inicio como veremos en el siguiente apartado.

Otro aspecto a señalar es la fecha de entrega del proyecto que se ha visto pospuesta en el tiempo considerablemente. La fecha prevista para la finalización era el 13 de julio de 2011, cosa que no se ha cumplido. Si bien es cierto que gran parte del proyecto se finalizó para antes de esa fecha, la documentación y pequeñas mejoras no fueron acabadas para esa fecha.

Por este motivo y otros asuntos personales se ha alargado durante un curso más la fecha de entrega del proyecto, fijándola para el mes de julio de 2012.



5.1.1. Análisis de las horas realizadas en el proyecto

Como se ha comentado anteriormente no se ha cumplido con el número de horas estimadas al inicio del proyecto. En un principio se estimaron 387 horas de trabajo, pero estas previsiones se han ido alargando hasta llegar a las 587 horas.

En la Fig. 26 se puede ver una gráfica con la comparación de horas estimadas y las reales de las tareas principales. En cada tarea se puede observar un aumento de horas respecto a lo planificado, pero donde realmente hay un aumento significativo es en las tareas de diseño y pruebas.

Por otro lado, este aumento también es debido a algunas mejoras realizadas que no estaban en la planificación inicial. La tarea de documentación es otra de las tareas que también ha requerido un mayor esfuerzo en horas.

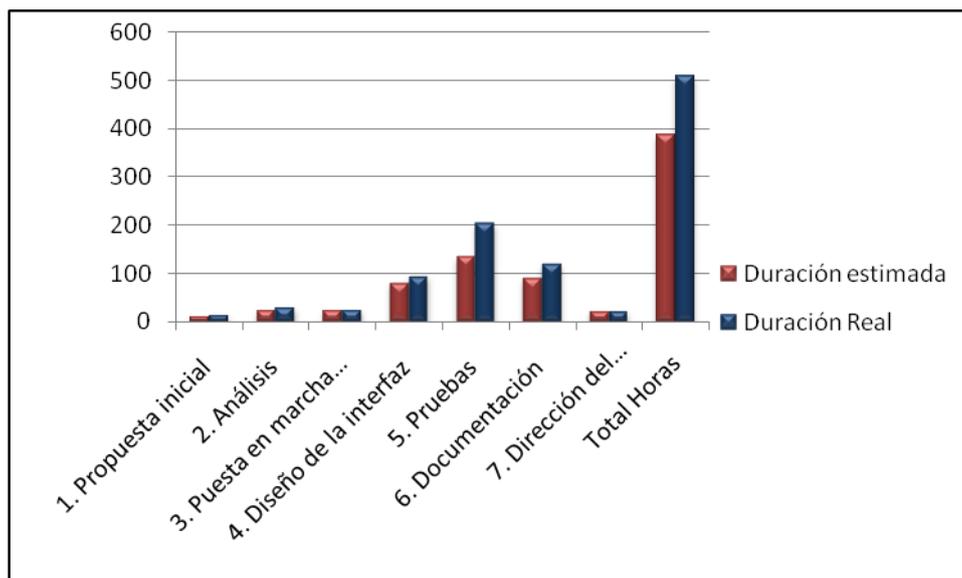


Fig. 26 Grafico con la comparación de horas de las tareas principales

En la siguiente tabla se puede ver con más detalle la comparación de horas de cada tarea:



Sistema de entrenamiento para un brazo articulado portátil

TAREA PRINCIPAL	SUBTAREA	DURACIÓN ESTIMADA	DURACIÓN REAL
1. Propuesta inicial (Documento de objetivos del proyecto)		12 horas	14 horas
2. Análisis	Documentación del brazo articulado	6 horas	12 horas
	Literatura sobre interacción persona robot	14 horas	14 horas
	Estudio sobre los dispositivos de interacción	4 horas	4 horas
3. Puesta en marcha e instalación del hardware y software	Instalación del brazo articulado	3 horas	2 horas
	Conexión al ordenador e instalación de los drivers	3 horas	3 horas
	Instalación de la interfaz básica	6 horas	3 horas
	Pruebas de control del brazo articulado	2 horas	2 horas
	Instalación de los dispositivos de interacción avanzados	4 horas	4 horas
4. Diseño de la interfaz	Elección del entorno de desarrollo y librerías a usar	6 horas	12 horas
	Propuesta de interfaz combinada avanzada	30 horas	40 horas
	Desarrollo del prototipo	50 horas	55 horas
5. Pruebas	Pruebas del prototipo	15 horas	15 horas
	Analizar resultados	10 horas	10 horas
	<u>Corrección e identificación de mejoras</u>		
	Nueva propuesta de interfaz gráfica de usuario	14 horas	20 horas
	Desarrollo del prototipo	40 horas	60 horas
	Desarrollo de un sistema de entrenamiento	32 horas	50 horas
6. Documentación	Nuevas pruebas	16 horas	16 horas
	Documentación	8 horas	8 horas
	Mejoras en el sistema de entrenamiento		25 horas
	Redacción de la memoria del proyecto	80 horas	100 horas
7. Dirección del proyecto, control de calidad y seguimiento	Manual de usuario	10 horas	20 horas
		22 horas	22 horas
		387 horas	511 horas

Tabla 12 Tabla detallada con la comparación de horas de cada tarea



5.1.2. Historia del proyecto

El proyecto se ha realizado durante dos cursos académicos, desde enero de 2011 hasta junio de 2012. La mayor parte del trabajo se ha realizado de enero a junio de 2011. A partir de esta fecha hubo un parón debido a motivos personales, y hasta abril de 2012 no se reanudó de nuevo. Éste es el resumen del trabajo realizado cada mes en el transcurso del proyecto:

- **Enero 2011:** Primeras reuniones en las que se definen los objetivos principales del proyecto.
- **Febrero 2011:** Se consulta documentación sobre la robótica enfocada a la rehabilitación, asistencia y compensación funcional de personas con discapacidades motoras. A finales del mes se comienza con la instalación de los drivers y hacer pruebas con la interfaz básica del brazo articulado.
- **Marzo 2011:** Durante este mes se elige el entorno de desarrollo y las librerías a utilizar. Se empieza con el desarrollo de la primera interfaz que hace uso de un *joystick*.
- **Abril 2011:** Se desarrolla una interfaz gráfica para controlar el brazo articulado y se realizan pruebas de las dos interfaces.
- **Mayo 2011:** En una reunión con el director se establece un nuevo objetivo, dotar de un sistema de entrenamiento al brazo articulado. Se empiezan a adaptar las interfaces desarrolladas para realizar el sistema de entrenamiento.
- **Junio 2011:** En este mes se mejoran la interfaz gráfica y la que hace uso del *joystick* para la integración en el sistema de entrenamiento. Se acuerdan con el director pequeñas mejoras de las interfaces.
- **Abril, Mayo y Junio 2012:** Los tres meses se dedican a redactar la memoria del proyecto. También se terminan de desarrollar pequeñas mejoras en las interfaces.



6. Conclusiones y líneas de trabajo futuras

En este apartado se van explicar las conclusiones a las que se ha llegado tras analizar el cumplimiento de los objetivos marcados en el proyecto. Se incluyen líneas de trabajo futuras basándose en este proyecto y recomendaciones sobre ampliación o continuación del proyecto.

6.1. Conclusiones

Desde hace tiempo la robótica está muy integrada en los procesos industriales, pero como hemos visto su incorporación en la vida cotidiana también está muy presente con robots como el iArm. Con este tipo de robots, se puede conseguir aprovechar su gran funcionalidad para el desarrollo de interfaces de diversa índole para personas con discapacidad motora.

Cuando se crean interfaces es importante tener en cuenta quienes van a ser los usuarios finales, más si se trata de personas con discapacidad motora. El sistema de entrenamiento que hemos creado está dirigido a este tipo de personas. Por lo que ha sido muy importante crear un sistema que se adapte al usuario y no al revés, dejando que sea él quien elija la configuración e interfaz que se adapte mejor a sus habilidades.

Este sistema de entrenamiento es un punto de partida, pero en el que queda pendiente el estudio de otras tecnologías que pueden ser útiles para ampliar el sistema. Con el uso de estas nuevas tecnologías se puede ampliar el número de interfaces o completar las existentes. Es importante disponer de distintas interfaces fáciles de usar e intuitivas debido a las distintas discapacidades y necesidades específicas que requiere cada usuario. Por lo tanto si el usuario tiene un sistema amplio podrá elegir cuál es la configuración e interfaz que más se adapta a sus habilidades.

6.2. Líneas de trabajo futuras

Durante la elaboración del proyecto, se han ido redefiniendo algunos de los objetivos originales y se han desechado algunas ideas por falta de tiempo para llevarlas a cabo. También se ha tenido en cuenta que se podría usar este



proyecto final de carrera como punto de partida de otros más complejos. Partiendo de este proyecto han surgido las ideas para los siguientes trabajos:

- **Realizar pruebas con el usuario final:** Las pruebas que se han realizado durante el proyecto no han involucrado al usuario final. En proyectos de este tipo es interesante involucrar al usuario final en el diseño y evaluación.
- **Estudiar el uso de reconocimiento de voz:** Estudiar e investigar la posibilidad de incorporar en el sistema de entrenamiento el reconocimiento por voz. La posibilidad de mover el brazo articulado mediante instrucciones por voz, haría que personas que no pudiesen mover los brazos tuviesen una alternativa para mover el brazo articulado.
- **Estudio de otros dispositivos de interacción persona-computador:**
Buscar otros dispositivos con los que crear nuevas interfaces para controlar el brazo articulado. Uso de otros dispositivos: reconocimiento de voz, pantallas táctiles, tablets, BCI (Brain-Computer Interfaces, etc. u otro tipos de dispositivos.
Para personas que apenas tienen movilidad, una solución para poder mover el brazo articulado es la posibilidad de hacerlo mediante las ondas cerebrales. Esto sería un gran avance para estas personas. A día de hoy, no existen versiones comerciales que ofrezcan este tipo de soluciones, pero sí que hay diversas instituciones desarrollando diversos prototipos:
 - Científicos del Instituto Federal de Tecnología en Lausanne han conseguido interpretar las órdenes mentales de una persona para controlar un robot mediante el uso de una gorra con electrodos⁸.
 - Científicos de la universidad de Brown han conseguido algo similar. Mediante una interfaz conectada a un sensor en el cerebro de una persona se recogen las señales eléctricas que se generan en las

⁸ <http://www.neoteo.com/robot-controlado-con-la-mente>



neuronas de la corteza motora, e interpretan las órdenes para controlar los movimientos de un brazo articulado con una enorme precisión⁹.

- **Investigar soluciones basadas en la visión artificial:** Muchas instituciones están investigando este tipo de soluciones. Este proyecto podría servir para investigar en nuevas interfaces basadas en la visión artificial.
- **Actualizar este proyecto fin de carrera:** se pueden realizar ciertas mejoras en el sistema de entrenamiento:
 - Aunque no hay restricción en el tipo de *joystick* a usar, dependiendo de *joystick* que se conecte adaptar la interfaz.
 - Integrar las dos interfaces en una única aplicación.

⁹ <http://www.neoteo.com/paciente-tetraplejica-controla-un-brazo-robotico-c>





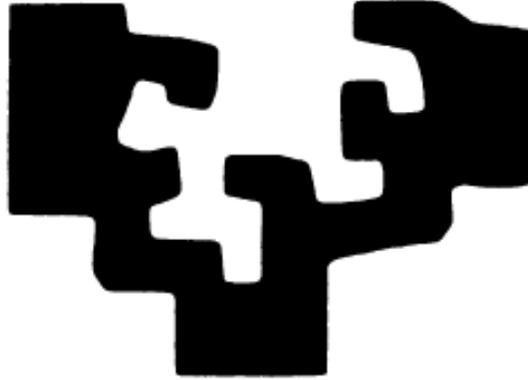
7. Bibliografía

- Kevin D. Edwards (University of South Florida). Design, construction and testing of a wheelchairmounted robotic arm. (2005)
- B J F Driessen, H G Evers and J A v Woerden. MANUS--a wheelchair-mounted rehabilitation robot: <http://pih.sagepub.com/content/215/3/285>
- Katherine Tsui and Holly Yanco(University of Massachusetts Lowell). Development and Evaluation of a Flexible Interface for a Wheelchair Mounted Robotic Arm
- Katherine M. Tsui and Holly A. Yanco. Survey of Domain-Specific Performance Measures in Assistive Robotic Technology.
- Katherine M. Tsui, Kareem Abu-Zahra, Renato Casipe, Jason M'Sadoques, and Jill L. Drury. Developing Heuristics for Assistive Robotics.
- Exact Dynamics. Assistive robotic manipulator: <http://www.exactdynamics.nl/site/>
- iArm Transparant Mode ESD CAN-USB Mini-interface
- iArm Instruction Manual
- Librería Qt: <http://www.zonaqt.com/>
- Librería SDL: <http://www.libsdl.org/>
- Qt Creator: <http://qt.nokia.com/products/developer-tools>.



- C++: <http://www.cplusplus.com/>
- MinGW: <http://www.mingw.org/>
- SDL Haptic: <http://wiki.libsdl.org/moin.cgi/CategoryForceFeedback>

eman ta zabal zazu



universidad
del país vasco

euskal herriko
unibertsitatea

Facultad de Informática Informatika Fakultatea

Sistema de entrenamiento para un brazo
articulado portátil

ANEXO I

Manual del Sistema de Entrenamiento Parte I :
Uso de una interfaz gráfica de usuario



INDICE

1. Introducción	1
2. Resumen funcionalidad interfaz gráfica	2
3. iArm.....	3
4. Estado actual del iArm	4
5. Movimientos automáticos	5
6. Movimientos cartesianos	7
7. Movimientos de la pinza (Orientación).....	8
8. Movimientos articulares	9
9. Movimientos del elevador	10
10. Abrir/Cerrar pinza	10
11. Cambiar velocidad	10



**Anexo I. Manual del Sistema de Entrenamiento Parte I : Uso de una
interfaz gráfica de usuario**



1. Introducción

La parte I del sistema de entrenamiento está basada en una interfaz gráfica de usuario. Mediante esta interfaz el usuario podrá controlar el brazo articulado iArm de una manera sencilla e intuitiva.

Con esta interfaz el usuario tiene a primera vista toda la funcionalidad para el control del brazo articulado. Para moverse sobre la interfaz gráfica es necesario un dispositivo de entrada que permita controlar el cursor sobre la interfaz. Para un usuario que esté habituado a mover el cursor en un ordenador, el uso de esta interfaz le resultará muy sencillo.

A parte de poder ser manejada mediante dispositivos de entrada que permiten controlar el cursor, esta interfaz ha sido diseñada para poder ser usada en pantallas táctiles. La disposición de los botones y otros elementos están pensados para poder ser accesibles en pantallas táctiles y que al usuario le resulte fácil. En este documento se va tratar en términos de “hacer clic” a la hora de realizar una acción sobre la interfaz, aunque se entiende que en una pantalla táctil las acciones a realizar sobre la interfaz son las mismas, pero en vez de clicar se pulsaría sobre los elementos de ella.

Una ventaja que nos ofrece esta interfaz, es que es muy visual y el usuario sabe que movimientos son los que quiere transmitir al brazo articulado sin tener la necesidad de aprenderse una configuración o la disposición de los elementos en pantalla para el manejo del robot.



2. Resumen funcionalidad interfaz gráfica

En la Fig.1 se puede ver de un vistazo toda la funcionalidad del brazo articulado a la que podemos acceder desde la interfaz gráfica.

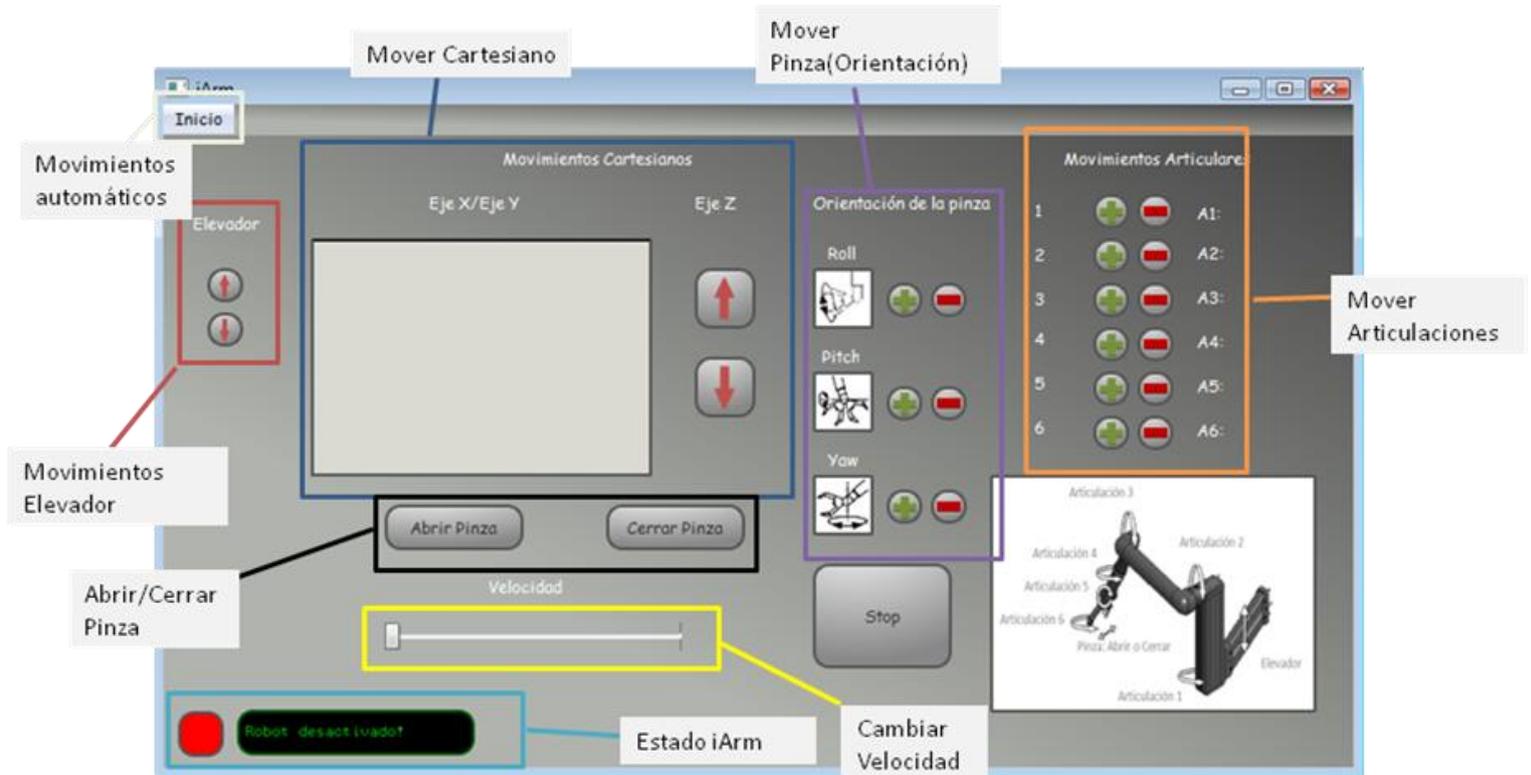


Fig. 1 Resumen funcionalidad de la interfaz gráfica

Resumen de las acciones que se pueden hacer sobre el iArm:

- Estado actual del iArm.
- Movimientos automáticos.
- Movimientos cartesianos.
- Movimientos de la pinza (Orientación).
- Movimientos articulares.
- Movimientos del elevador.
- Abrir o Cerrar pinza.
- Cambiar la velocidad.



3. iArm

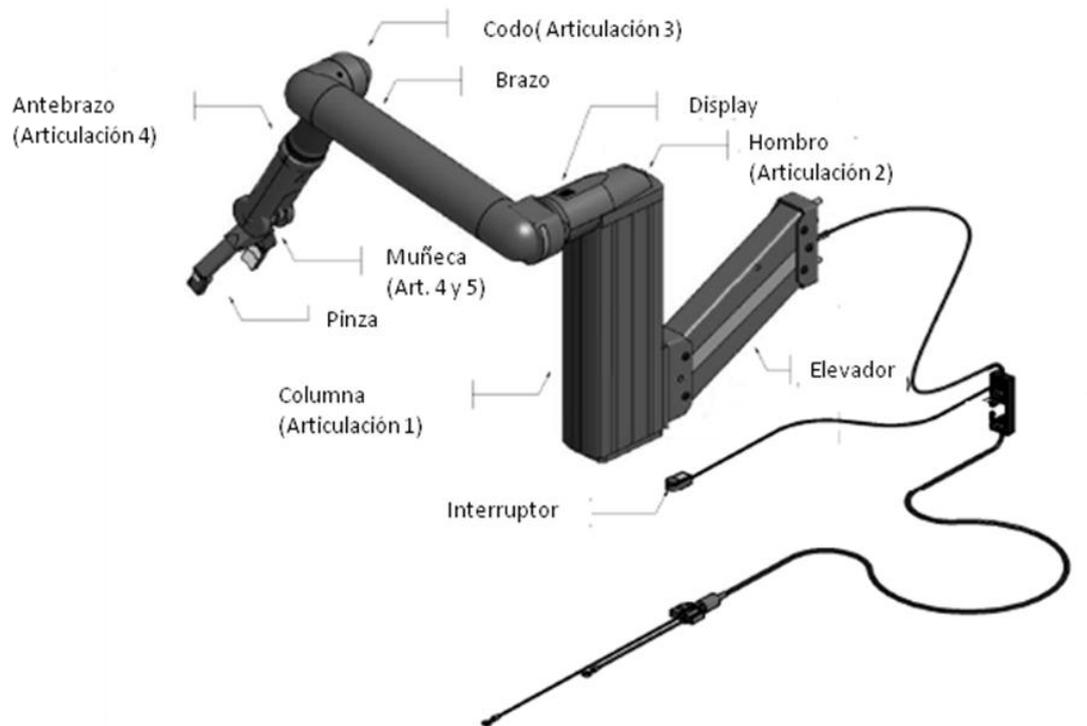


Fig. 2 Parte del iArm

Los diferentes componentes del iArm tienen nombres que coinciden con los nombres del brazo humano, por ejemplo, brazo, hombro, muñeca. Las articulaciones están numeradas de la 1 a la 6.

El iArm también tiene un display donde se muestran distintos mensajes de aviso y de error.

También dispone de un interruptor que sirve para encender y apagar el iArm. Se puede poner en modo de espera “standby”.



4. Estado actual del iArm

En la parte inferior izquierda de la interfaz gráfica se muestra el estado en todo momento del iArm.

Cuando el brazo articulado esté apagado o no tenga conexión se mostrará el siguiente mensaje y no se podrá hacer uso de la interfaz:

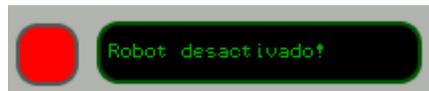


Fig. 3 Estado desactivado

Para empezar a controlar el brazo articulado mediante la interfaz se tiene que encender el robot y establecer la conexión. Pulsando el interruptor del iArm durante aproximadamente un segundo se activa el iArm. Tras unos segundos de arranque la interfaz estará disponible. En el *display* del iArm se mostrará la imagen de la izquierda y en la interfaz la imagen de la derecha (Fig. 4). En este estado se podrá empezar a controlar y transmitir los movimientos al brazo articulado.



Fig. 4 Estado activo

Cuando se realice algún movimiento automático como plegar o desplegar el robot y éste termine, se mostrará un mensaje como el de la Fig.5 durante unos segundos acompañado de un pitido para avisar al usuario de que el movimiento automático se ha terminado con éxito y que se pueden realizar otros movimientos.

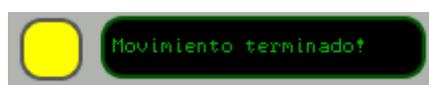


Fig. 5 Movimiento terminado



Cuando el usuario intente realizar un movimiento que el robot no pueda realizar, por ejemplo movimientos en puntos singulares, se mostrará un mensaje como el de la Fig. 6 acompañado de un pitido.

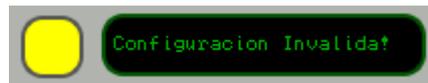


Fig. 6 Configuración inválida

5. Movimientos automáticos

Son movimientos que el iArm realiza automáticamente. Se seleccionan en la parte izquierda superior de la interfaz. Pulsando en la pestaña Inicio se nos muestran los tres movimientos automáticos disponibles que hay.

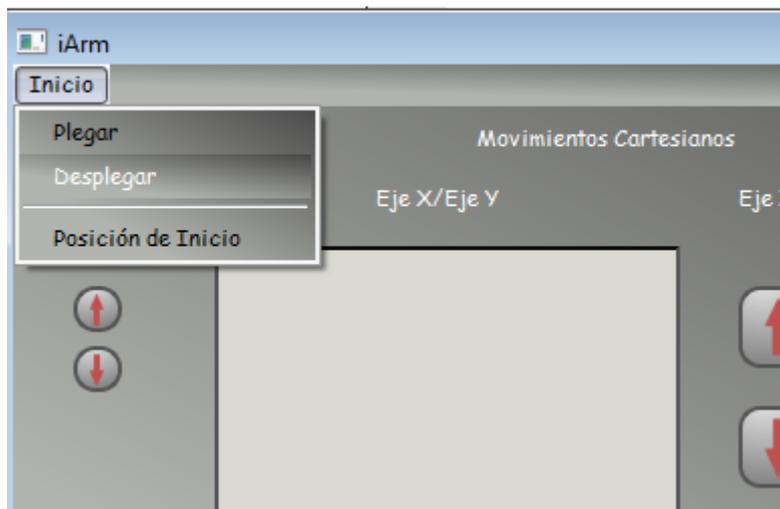


Fig. 7 Movimientos Automáticos

Mediante las opciones Plegar o Desplegar se puede recoger o desplegar el robot de una manera rápida como se muestra en la Fig. 8 de tal modo que se pueda empezar a controlar el robot con otros movimientos o terminar de hacerlo. No siempre se podrán realizar estos movimientos, en ciertas configuraciones articulares el iArm no puede hacer estos movimientos automáticos y se mostrará un mensaje de configuración inválida.



Fig. 8 A la izquierda robot plegado. A la derecha robot desplegado.

Cuando los movimientos de plegar o desplegar no se pueden realizar, se recomienda usar la opción de posición de inicio. Mediante esta opción el iArm se situará en la misma posición que la de despliegue. Desde esta posición el robot se podrá plegar o hacer otros movimientos. Con el robot plegado se recomienda no usar esta opción ya que se puede dañar el robot.

Para hacer los movimientos automáticos se recomienda que haya espacio alrededor del iArm, puesto que se trata de movimientos automáticos.

Si el iArm está realizando un movimiento automático y el usuario intenta realizar cualquier otro movimiento, se dejará de realizar el movimiento automático para realizar el último movimiento.

Para parar de hacer un movimiento automático la interfaz dispone de un botón de parada (Fig. 9). Con este botón se para cualquier movimiento que esté haciendo el iArm, aunque es en los movimientos automáticos donde tiene más sentido su utilización.



Fig. 9 Botón de parada



6. Movimientos cartesianos

Mediante estos movimientos se mueve el punto de centro de herramienta (TCP) de la ubicación en la que está con respecto a la base en múltiples direcciones (sobre las coordenadas x , y , z). Es decir, el usuario puede mover hacia los lados, adelante o atrás, arriba abajo la pinza. El (TCP) estará situado normalmente en la pinza.

Para transmitir los movimientos cartesianos al iArm tenemos dos elementos, un recuadro que simula un “touchpad” para las coordenadas X , Y , y los botones arriba, abajo para la coordenada Z .

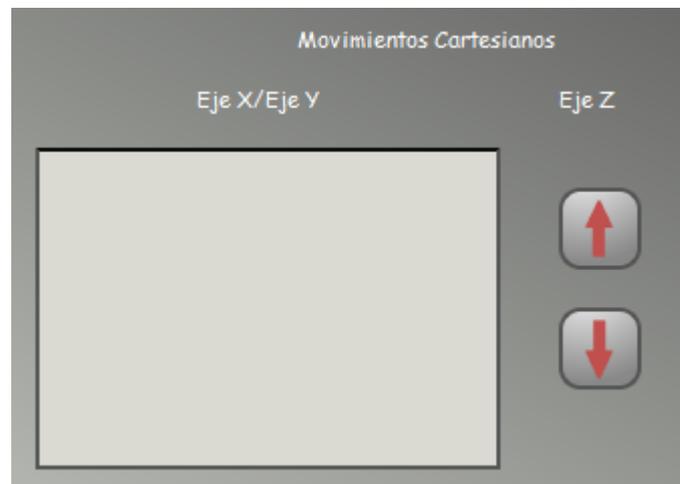


Fig. 10 Movimientos Cartesianos

Para utilizar el recuadro gris, que simula un *touchpad*, el usuario tendrá que clicar dentro del recuadro y moverse hacia adelante, atrás, derecha o izquierda o en diagonales con el botón izquierdo clicado. Según el movimiento que se haga dentro de este recuadro, el brazo articulado moverá la pinza sobre las coordenadas X e Y . Este recuadro facilita al usuario mover la pinza en el espacio y simplifica el número de acciones a realizar sobre la interfaz ya que se puede mover la pinza sobre las coordenadas X e Y simultáneamente.

Para el eje Z , se dispone de dos botones en los que clicando sobre ellos se moverá la pinza en la coordenada Z .

Si el usuario intenta mover la pinza cuando se acerca un punto singular interno no podrá hacerlo a la velocidad especificada, o si lo alcanza el iArm perderá uno o más grados de libertad y tendrá que reducir su velocidad angular



y linear durante el acercamiento para garantizar una trayectoria recta durante el movimiento.

Además, si el usuario mueve la pinza donde se alcanza una configuración singular en los límites del espacio de trabajo del robot, se generará un mensaje de aviso de configuración inválida y no se dejará realizar el movimiento.

7. Movimientos de la pinza (Orientación)

Mediante estos movimientos el usuario podrá orientar la pinza en la posición deseada. Hay tres movimientos que el usuario podrá realizar para orientar la pinza:

- Roll (En castellano Alabeo): rotación respecto al eje.



Fig. 11 Roll

- Pitch (En castellano Cabeceo): es una inclinación sobre el eje.

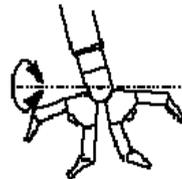


Fig. 12 Pitch

- Yaw (En castellano Guiñada): rotación intrínseca alrededor del eje vertical perpendicular a la pinza.

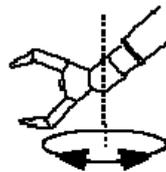


Fig. 13 Yaw



En la Fig.14 podemos ver como para cada movimiento de orientación tenemos dos botones. Mediante estos botones se realizarán los movimientos de orientación de cada movimiento en sentido positivo o negativo.



Fig. 14 Movimientos de la pinza

8. Movimientos articulares

El iArm dispone de 6 articulaciones que pueden ser movidas individualmente.

Cada una de las articulaciones se pueden mover mediante los botones “+” y “-” en sentido positivo y negativo. Además a la derecha de los botones de cada articulación se muestra la posición angular de cada articulación.



Fig. 15 Movimientos Articulares



9. Movimientos del elevador

Se puede mover el elevador del iArm hacia arriba o abajo mediante los botones de la interfaz.



Fig. 16 Elevador

10. Abrir/Cerrar pinza

Clicando los botones Abrir pinza o Cerrar Pinza se puede abrir y cerrar la pinza. Cuando el usuario cierra la pinza, se aplica una tensión en el resorte. Si la pinza toca un objeto, el usuario puede seguir cerrando la pinza incrementando su fuerza. La pinza tiene un sistema que garantiza que los objetos son manipulados con una compresión de tres puntos.



Fig. 17 Abrir/Cerrar Pinza

11. Cambiar velocidad

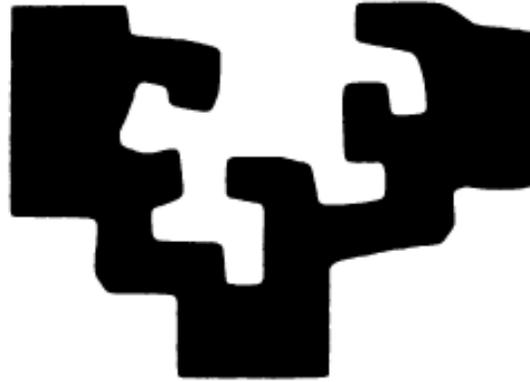
El usuario puede cambiar la velocidad de los movimientos del iArm mediante la barra de desplazamiento.



Fig. 18 Cambiar velocidad

El cambio de velocidad afecta a todos los movimientos excepto a los movimientos automáticos que se mueven a una velocidad por defecto.

eman ta zabal zazu



universidad
del país vasco

euskal herriko
unibertsitatea

Facultad de Informática Informatika Fakultatea

Sistema de entrenamiento para un brazo
articulado portátil

ANEXO II

Manual del Sistema de Entrenamiento Parte II :
Uso del joystick



INDICE

1. Introducción	1
2. iArm.....	1
3. Joystick.....	2
4. Configuración del joystick	3
4.1. Guardar/Abrir configuración del joystick	4
4.2. Configuración de los movimientos	4
4.3. Ver configuración	8
4.4. Activar joystick	8
5. Movimientos del iArm.....	9
5.1. Estado actual del iArm	9
5.2. Movimientos automáticos	10
5.3. Movimientos cartesianos	11
5.4. Movimientos de la pinza (Orientación).....	12
5.5. Movimientos articulares	12
5.6. Movimientos del elevador	12
5.7. Abrir/Cerrar pinza.....	13
5.8. Cambiar velocidad	13



Anexo II. Manual del Sistema de Entrenamiento Parte II : Uso del joystick



1. Introducción

La parte II del sistema de entrenamiento está basada en el uso de un *joystick*. Mediante esta interfaz el usuario podrá controlar el brazo articulado iArm de una manera sencilla e intuitiva a través del *joystick*.

Esta aplicación está diferenciada en dos partes. En la primera parte el usuario tendrá que configurar el *joystick* asignándole los movimientos que desee mediante una interfaz gráfica, y la segunda parte consistirá en controlar el iArm mediante el dispositivo.

Con esta interfaz, el usuario tiene la opción de configurar el *joystick* según sus preferencias y habilidades para después controlar el brazo articulado.

2. iArm

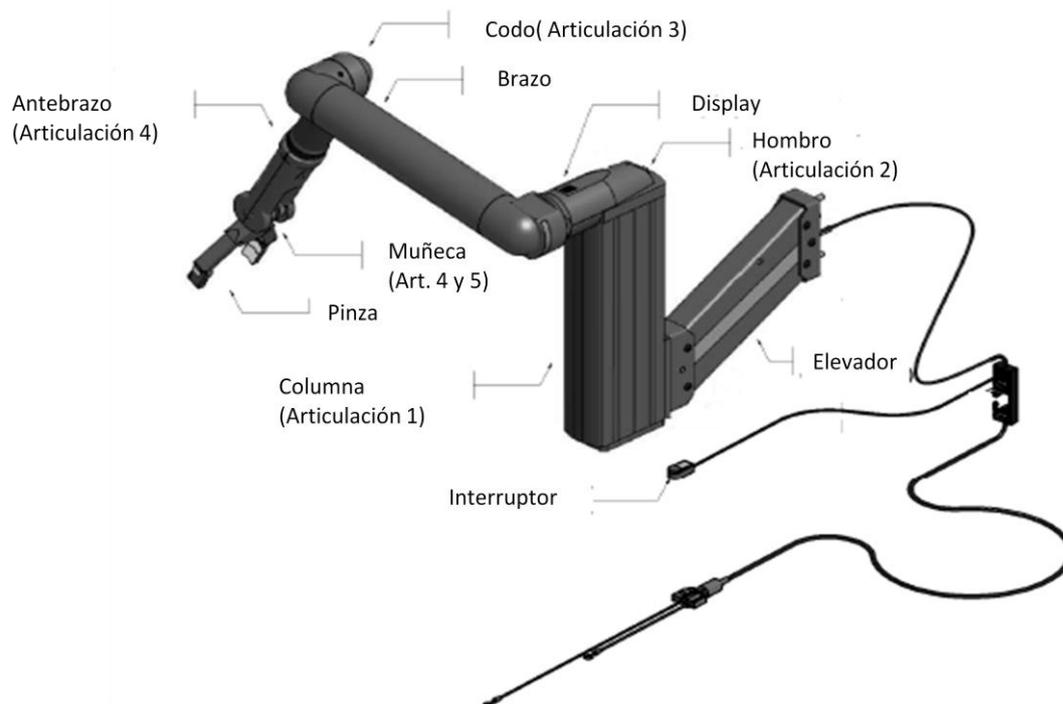


Fig. 1 Parte del iArm

Los diferentes componentes del iArm tienen nombres que coinciden con los nombres del brazo humano, por ejemplo, brazo, hombro, muñeca. Las articulaciones están enumeradas de la 1 a la 6.



El iArm también tiene un *display* donde se muestran distintos mensajes de aviso y de error.

También dispone de un interruptor que sirve para encender y apagar el iArm. Se puede poner en modo de espera “*standby*”.

3. Joystick

La aplicación admite cualquier tipo de *joystick* siempre que el sistema operativo bajo el que se esté ejecutando la aplicación sea capaz de reconocerlo. Pero si se dispone de un *joystick* con un número limitado de ejes o botones no se podrá realizar una configuración completa con la que poder controlar todos los movimientos del iArm.

Según el tipo de *joystick* que se conecte al PC, la aplicación reconocerá el número de botones, ejes y botones direccionales que tiene para adaptar la interfaz gráfica de usuario de configuración del *joystick*.

Es aconsejable que el *joystick* disponga de “*forcefeedback*”, ya que si se tiene, la aplicación hará uso de él y ofrecerá al usuario mayor funcionalidad a la hora de controlar el brazo articulado.

Tanto para hacer este manual como para la interfaz gráfica se ha tomado como referencia el *joystick* “Force 3D PRO” de la marca Logitech. Este modelo dispone de 12 botones, 3 ejes y un botón direccional que permite realizar al usuario una configuración completa para tener un control sobre todas las funcionalidades del iArm, además dispone de “*forcefeedback*”. Las imágenes utilizadas en la interfaz gráfica pertenecen a este modelo de *joystick* y han sido utilizadas para dar al usuario una idea de la configuración que están realizando, pero esto no limita que se pueda hacer uso de otros *joysticks*.



4. Configuración del joystick

Antes de empezar a controlar el iArm con el *joystick* el usuario podrá elegir la configuración que desee mediante la interfaz gráfica de usuario de la aplicación. En la Fig. 1 se puede ver la pantalla en la que el usuario podrá asignar a cada uno de los movimientos del iArm una acción del *joystick*.

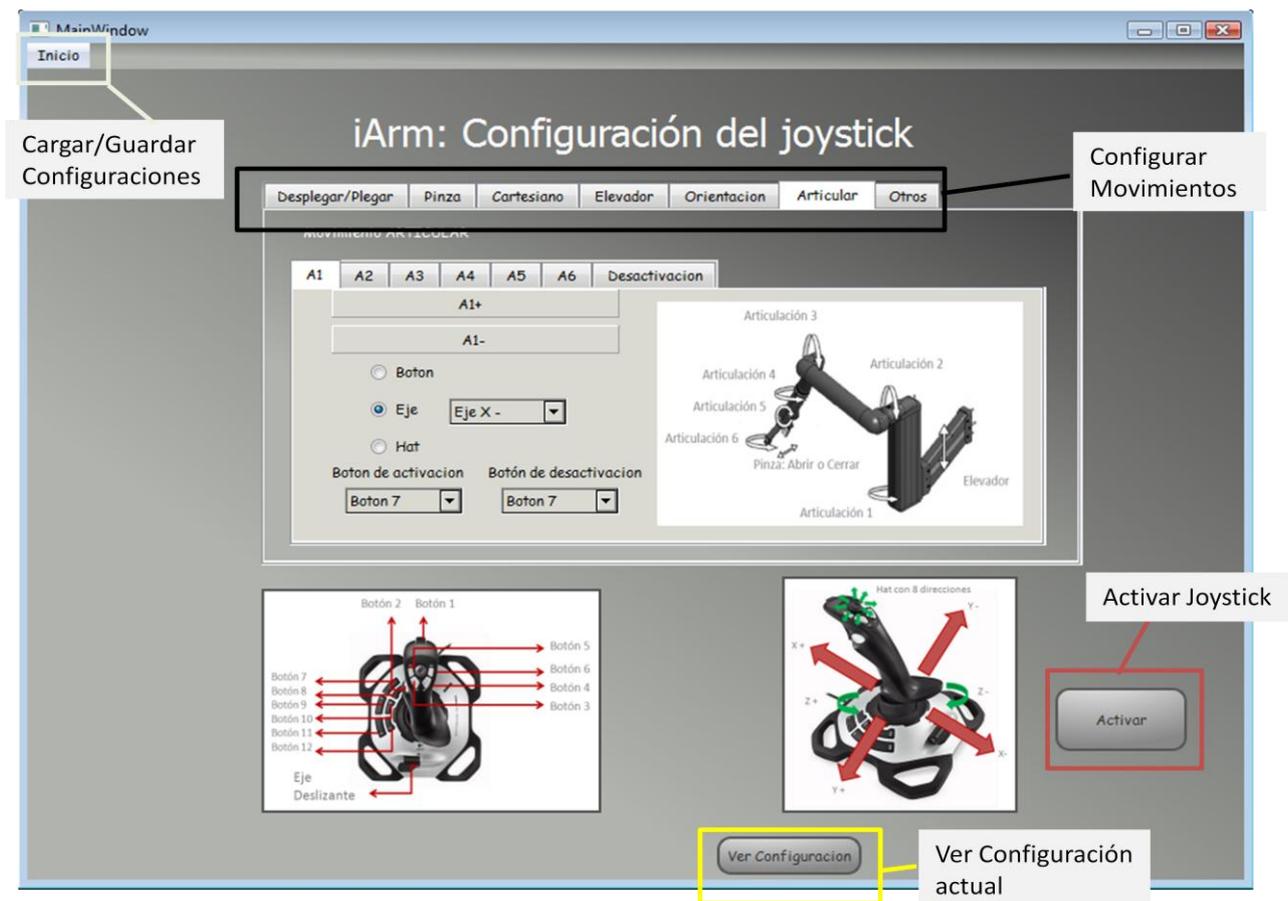


Fig. 2 Interfaz gráfica de usuario para la configuración del joystick

Resumen de acciones que se pueden realizar sobre la interfaz gráfica de usuario:

- Guardar o abrir una configuración del *joystick*.
- Pestañas para configurar cada uno de los movimientos del iArm.
- Ver la configuración del *joystick* actual.
- Activar el uso del *joystick* para controlar el iArm.



4.1. Guardar/Abrir configuración del joystick

La aplicación permite al usuario guardar o cargar una configuración para el *joystick*. Mediante la pestaña izquierda superior podemos acceder a las dos opciones.



Fig. 3 Guardar/Abrir configuración

Por defecto, la aplicación se carga con una configuración concreta. Pero el usuario puede abrir una configuración que haya guardado previamente. Para guardar la configuración elegirá una ubicación del sistema, donde se guardará en un archivo de texto y se podrá seleccionar posteriormente para abrir la configuración guardada.

4.2. Configuración de los movimientos

Para que el usuario entienda cómo configurar el *joystick*, la interfaz gráfica está organizada en pestañas por cada tipo de movimiento del brazo articulado que se puede controlar. Hay seis pestañas que pertenecen a seis grupos de movimientos y otra pestaña que contiene otros movimientos.



Fig. 4 Pestañas para cada uno de los movimientos

Los grupos de movimientos Desplegar/plegar, Pinza y Elevador contienen el movimiento propio y su opuesto. Es decir, para controlar la pinza podemos abrirla y su opuesto sería cerrarla. Por otro lado, tenemos los grupos de movimientos Articular, Orientación y Cartesiano que engloban más movimientos y sus opuestos. Por ejemplo, el grupo de movimientos cartesianos engloba los movimiento cartesianos en las coordenadas X, Y y Z y sus



opuestos. En la siguiente lista se muestran los grupos de movimientos y por cada uno de ellos a los movimientos que engloban junto con sus opuestos:

- **Desplegar/Plegar:**
 - Desplegar – Plegar
- **Pinza**
 - Abrir – Cerrar
- **Elevador**
 - Arriba – Abajo

Cartesiano

- **X:**
 - X+ – X-
- **Y:**
 - Y+ – Y-
- **Z:**
 - Z+ – Z-

Articular:

- **Articulación 1:**
 - A1+ – A1-
- **Articulación 2:**
 - A2+ – A2-
- **Articulación 3:**
 - A3+ – A3-
- **Articulación 4:**
 - A4+ – A4-
- **Articulación 5:**
 - A5+ – A5-
- **Articulación 6:**
 - A6+ – A6-

Orientación:

- **Pitch:**
 - Pitch + – Pitch-
- **Yaw:**
 - Yaw+ – Yaw-
- **Roll:**
 - Roll+ – Roll-

A cada uno de estos movimientos se le puede asignar una acción del *joystick*. Las acciones del *joystick* podrán ser sobre los botones, ejes o hat (botón direccional). El usuario tendrá que tener en cuenta no asignar la misma acción del *joystick* a distintos movimientos porque el sistema no lo impedirá.

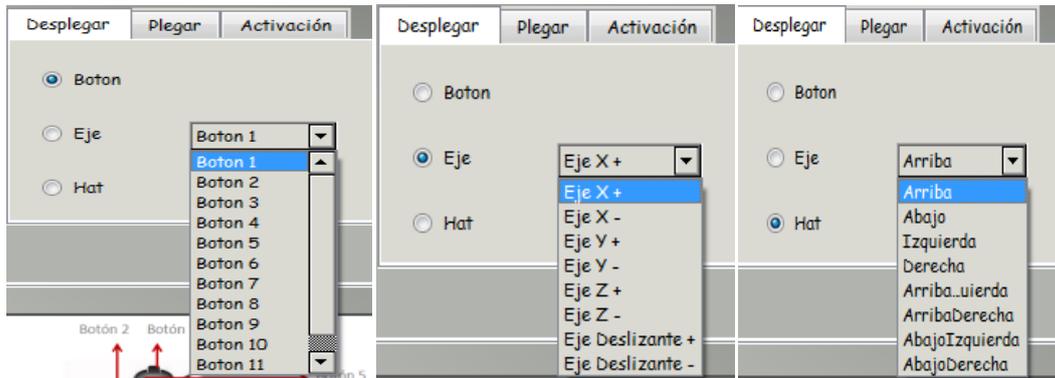


Fig. 5 Tipos de acciones del joystick

Para poder utilizar una misma acción del *joystick* para realizar los movimientos del robot tenemos la posibilidad de activarlos y desactivarlos. A cada uno de los movimientos se les podrá asignar un botón para activarlo y otro (o el mismo) para desactivarlos. También existe la posibilidad de no asignarle ninguno.

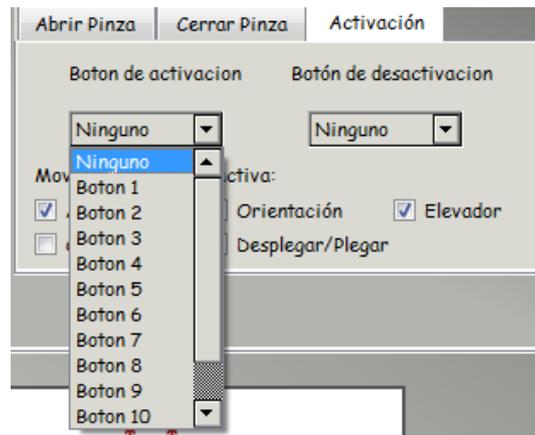


Fig. 6 Botones de activación y desactivación de los movimientos

La activación de un movimiento puede implicar la desactivación de otro. El usuario podrá configurar el *joystick* para que cuando se active un movimiento se desactiven otros grupos de movimientos. Esto permite usar las mismas acciones del *joystick* para controlar distintos movimientos del iArm.

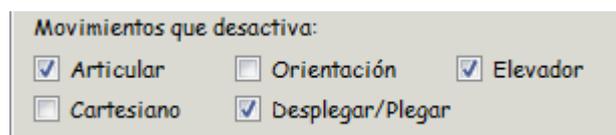


Fig. 7 Desactivación de movimientos



Si un usuario quisiese hacer dos movimientos distintos teniendo configurada la misma acción del *joystick* para ellos, el sistema solo haría uno de los dos movimientos dejando fuera de uso el otro. Para que el usuario pueda seleccionar el movimiento que quiere realizar, tendrá que activar el movimiento mediante el botón de activación o si no tiene botón de activación realizando la acción del *joystick* que tiene asignada. Esto implicará que si el usuario ha puesto que se desactive el otro movimiento, este se desactivará y se hará el movimiento deseado con la acción del *joystick* configurada.

Por ejemplo, si un usuario quiere abrir la pinza y mover el elevador hacia arriba con la acción del *joystick* del botón 1 para ambos movimientos, tendrá que seleccionar un botón de activación para uno de los dos movimientos y seleccionar que se desactive el otro movimiento. Así, si por ejemplo ha seleccionado que con el botón 2 activa los movimientos del elevador y que los movimientos del elevador desactivan los movimientos de la pinza, si pulsa el botón 2 del joystick activará los movimientos del elevador desactivando los de la pinza y con el botón 1 podrá mover el elevador hacia arriba.

Esto permite al usuario asignar cualquier tipo de acción sobre el *joystick* a cualquier movimiento del brazo articulado, con lo que tendrá un número amplio de opciones para configurar el *joystick* y elegir la configuración más acorde a sus características.

A parte de los movimientos mencionados, el usuario también podrá configurar las acciones del *joystick* para seleccionar la velocidad de los movimientos del iArm y para realizar el movimiento automático de posición de inicio, que sitúa la pinza del robot en la posición de despliegue.

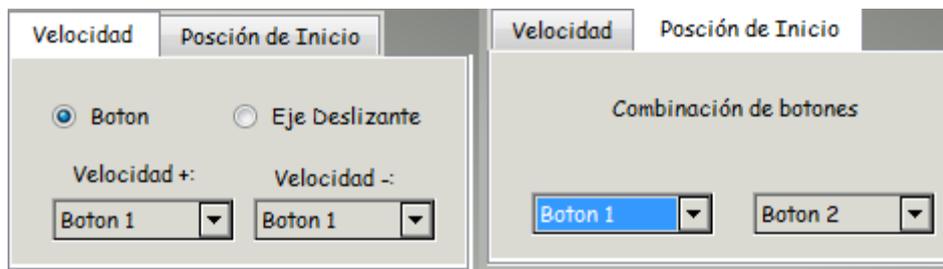


Fig. 8 Configuración de otros movimientos



Para estos movimientos no podrá utilizar cualquier acción del *joystick*. Para aumentar o reducir la velocidad de los movimientos del brazo articulado podrá utilizar dos botones o un eje deslizante y para realizar el movimiento de posición de inicio tendrá que seleccionar la combinación de dos botones.

4.3. Ver configuración

Mediante el botón “Ver configuración”, el usuario podrá ver de una forma visual un resumen de las acciones del *joystick* que ha asignado para controlar el brazo articulado.

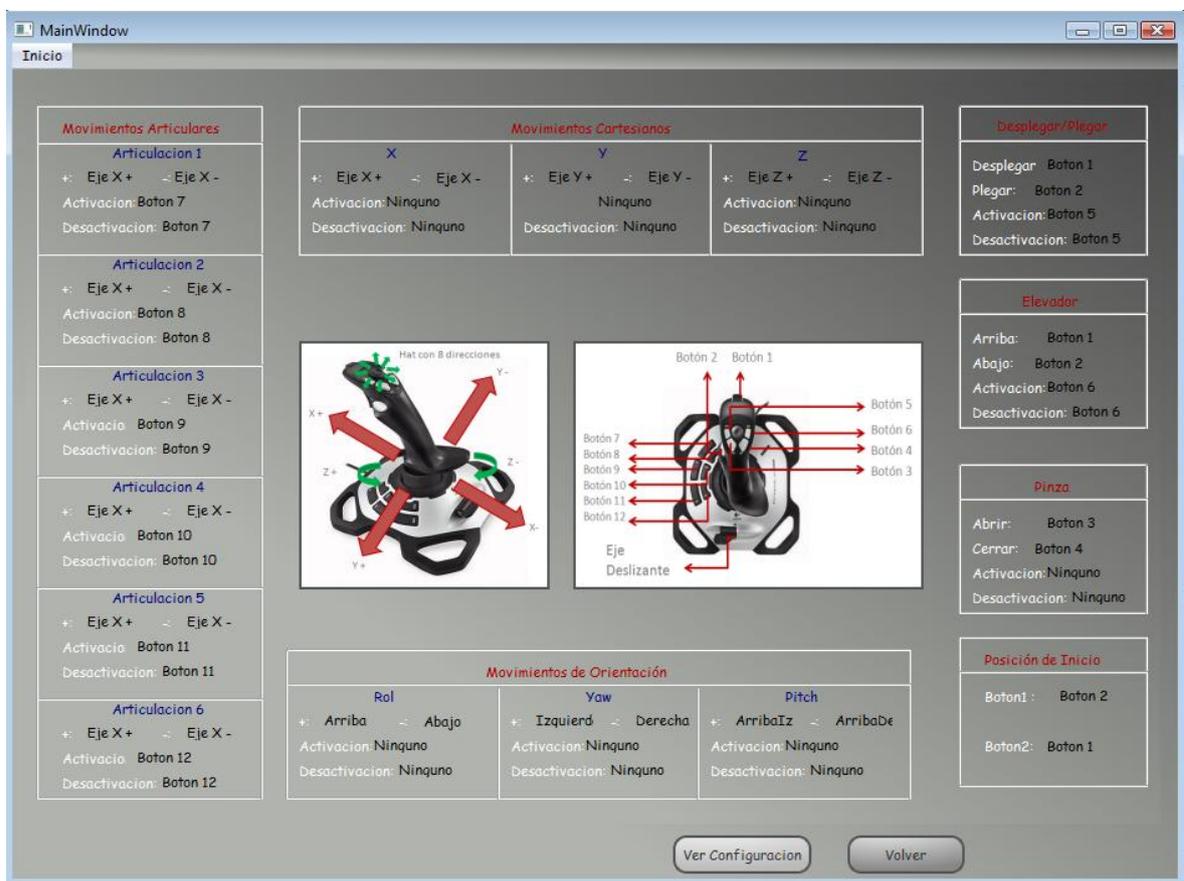


Fig. 9 Visualización de la configuración del joystick

4.4. Activar joystick

Una vez que el usuario haya configurado las acciones del *joystick*, mediante el botón “activar” podrá empezar a controlar el iArm. Se mostrará una pantalla con el resumen de la configuración seleccionada y el estado del iArm.



Fig. 10 Botón de activación

5. Movimientos del iArm

5.1. Estado actual del iArm

En la parte inferior izquierda de la interfaz gráfica se muestra el estado en todo momento del iArm.

Cuando el brazo articulado esté apagado o no tenga conexión se mostrará el siguiente mensaje y no se podrá hacer uso del *joystick*:

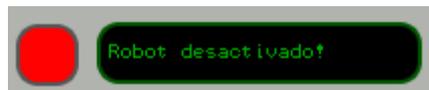


Fig. 11 Estado desactivado

Para empezar a controlar el brazo articulado mediante el *joystick* se tiene que encender el robot y establecer la conexión. Pulsando el interruptor del iArm durante aproximadamente un segundo se activa el iArm. Tras unos segundos de arranque habrá un pequeño movimiento de “forcefeedback” sobre los ejes del *joystick* y en el *display* del iArm se mostrará la imagen de la izquierda y en la interfaz la imagen de la derecha. En este estado se podrá empezar a controlar y transmitir los movimientos al brazo articulado.



Fig. 12 Estado activo

Cuando se realice algún movimiento automático como plegar o desplegar el robot y este termine se mostrará un mensaje como el de la Fig.14 durante unos segundos acompañado de un pitido y un pequeño movimiento en los ejes



del joystick para avisar al usuario de que el movimiento automático se ha terminado con éxito y que se pueden realizar otros movimientos.

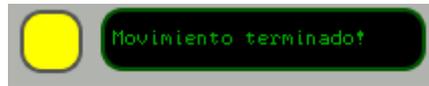


Fig. 13 Movimiento terminado

Cuando el usuario intente realizar un movimiento que el robot no pueda realizar, por ejemplo movimientos en puntos singulares, se mostrará un mensaje como el de la Fig. 6 acompañado de un pitido y la activación del “forcefeedback” del *joystick*.

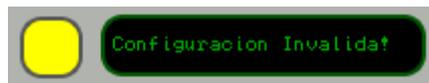


Fig. 14 Configuración inválida

5.2. *Movimientos automáticos*

Son movimientos que el iArm realiza automáticamente. Hay disponibles tres movimientos automáticos.

Mediante las opciones Plegar o Desplegar podremos recoger o desplegar el robot de una manera rápida como en la Fig. 15. No siempre se podrán realizar estos movimientos, en ciertas configuraciones articulares el iArm no puede hacer estos movimientos automáticos.



Fig. 15 A la izquierda robot Plegado. A la derecha robot desplegado.



Cuando los movimientos de plegar o desplegar no se pueden realizar, se recomienda usar la opción de posición de inicio. Mediante esta opción el iArm se situará en la misma que la de despliegue. Desde esta posición el robot se podrá plegar o hacer otros movimientos. Con el robot plegado se recomienda no usar esta opción ya que se puede dañar el robot.

Para hacer los movimientos automáticos se recomienda que haya espacio alrededor del iArm, puesto que se trata de movimientos automáticos.

5.3. Movimientos cartesianos

Mediante estos movimientos se mueve el punto de centro de herramienta (TCP) de la ubicación en la que está con respecto a la base en múltiples direcciones (sobre las coordenadas x, y, z). Es decir, el usuario puede mover hacia los lados, adelante o atrás, arriba abajo la pinza. El (TCP) está situado normalmente en la pinza.

El usuario puede configurar el *joystick* de tal manera que puede mover la pinza sobre el eje X, Y, Z simultáneamente reduciendo el número de acciones.

Si el usuario intenta mover la pinza cuando se acerca un punto singular interno no podrá hacerlo a la velocidad especificada, o si lo alcanza el iArm perderá uno o más grados de libertad y tendrá que reducir su velocidad angular y lineal durante el acercamiento para garantizar una trayectoria recta durante el movimiento. Además, el “*forcefeedback*” del *joystick* se activará avisando al usuario de que está cerca de un punto singular para que pueda corregir la posición.

Si el usuario mueve la pinza donde se alcanza una configuración singular en los límites del espacio de trabajo del robot se activará el “*forcefeedback*” y se generará un mensaje de aviso de configuración inválida no dejando realizar el movimiento.



5.4. *Movimientos de la pinza (Orientación)*

Mediante estos movimientos el usuario podrá orientar la pinza en la posición deseada. Hay tres movimientos que el usuario podrá realizar para orientar la pinza:

- Roll (En castellano Alabeo): Rotación respecto al eje.



Fig. 16 Roll

- Pitch (En castellano Cabeceo): Es una inclinación sobre el eje.



Fig. 17 Pitch

- Yaw (En castellano Guiñada): rotación intrínseca alrededor del eje vertical perpendicular a la pinza.

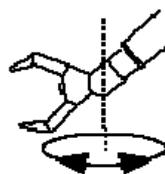


Fig. 18 Yaw

5.5. *Movimientos articulares*

El iArm dispone de 6 articulaciones que pueden ser movidas independientemente.

5.6. *Movimientos del elevador*

Se puede mover el elevador del iArm hacia arriba o abajo.



5.7. Abrir/Cerrar pinza

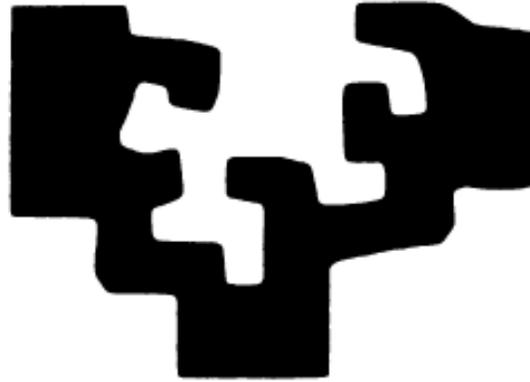
Clicando los botones Abrir pinza o Cerrar Pinza se puede abrir y cerrar la pinza. Cuando el usuario cierra la pinza, se aplica una tensión en el resorte. Si la pinza toca un objeto, el usuario puede seguir cerrando la pinza incrementando su fuerza. La pinza tiene un sistema que garantiza que los objetos son manipulados con una compresión de tres puntos.

5.8. Cambiar velocidad

El usuario puede cambiar la velocidad de los movimientos del iArm. El cambio de velocidad afecta a todos los movimientos excepto a los movimientos automáticos que se mueven a una velocidad por defecto.



eman ta zabal zazu



universidad
del país vasco

euskal herriko
unibertsitatea

Facultad de Informática Informatika Fakultatea

Sistema de entrenamiento para un brazo
articulado portátil

ANEXO III

Manual de instalación



INDICE

1. Introducción	1
2. Instalación del iArm	1
3. Instalación del entorno de desarrollo	1
3.1. MinGW	2
3.2. QT SDK	3
3.3. QT Creator	4
4. Librería iArm.....	5
5. Librería SDL.....	8





1. Introducción

Este manual pretende explicar cómo crear aplicaciones para el iArm. Se explica la instalación del iArm y del entorno de desarrollo Qt creator que forma parte del Qt SDK. Por último, se explica cómo usar las librerías para el iArm y las librerías SDL para usar el *joystick*. El manual está pensado para usarse con Windows, aunque las herramientas pueden ser usadas bajo el sistema operativo Linux.

2. Instalación del iArm

Hay que instalar los drivers del ESD ELECTRONICS USB-CAN-Interface. Para ello hay que conectar el cable USB-CAN-Interface al PC. Si el sistema no reconoce el USB-CAN-Interface habrá que instalar los drivers desde el CD.

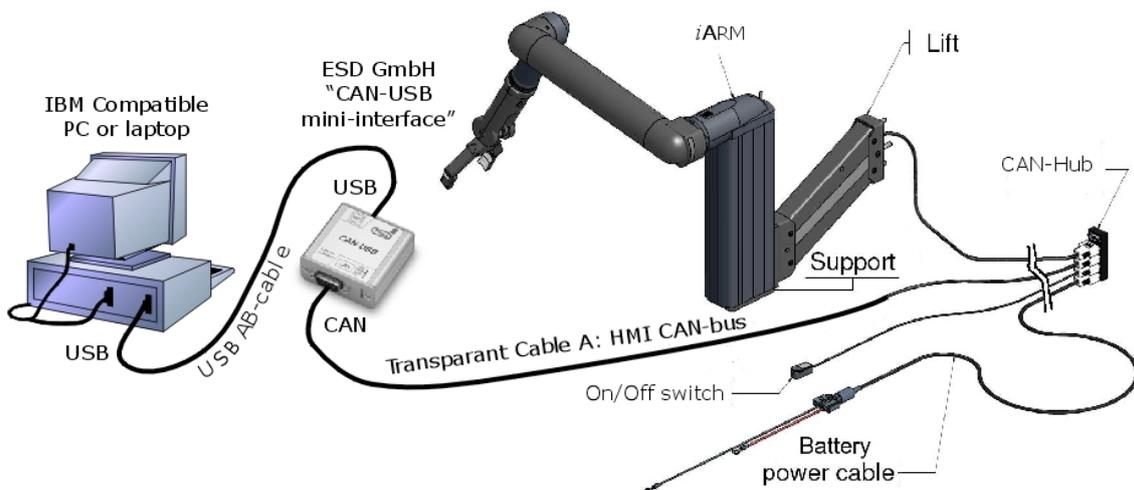


Fig. 1 Conexiones iArm

EL iArm hay que conectarlo mediante el "Transparent Cable A" al USB-CAN-Interface.

3. Instalación del entorno de desarrollo

Para el desarrollo de las aplicaciones y programas para el iArm se ha utilizado el entorno de desarrollo Qt Creator. Qt Creator es un IDE para el



desarrollo de aplicaciones con las bibliotecas Qt, usadas para aplicaciones con interfaces gráficas de usuario.

Para compilar las aplicaciones, será necesario un compilador del lenguaje de programación C++, en este caso, el MinGW para windows.

3.1. MinGW

Desde el siguiente enlace podremos descargarnos la última versión para Windows del compilador MinGW: <http://sourceforge.net/projects/mingw/>



Fig. 2 Descarga MinGW

Instalaremos el ejecutable:

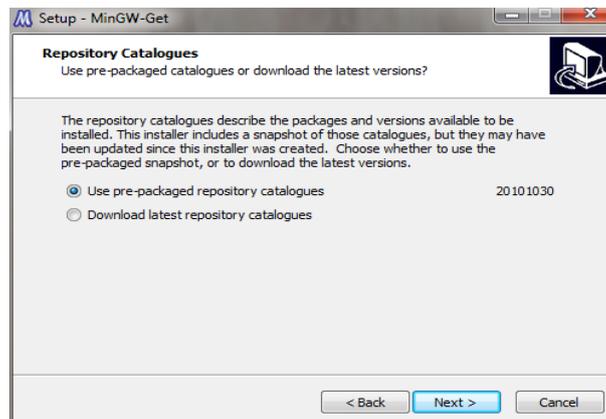


Fig. 3 Instalación MinGW

Lo instalamos en el directorio C:\MinGW u otro opcional:

En las opciones adicionales vamos a seleccionar lo siguiente:

- C Compiler (que viene por defecto).
- C++ Compiler.
- MSYS Basic System.

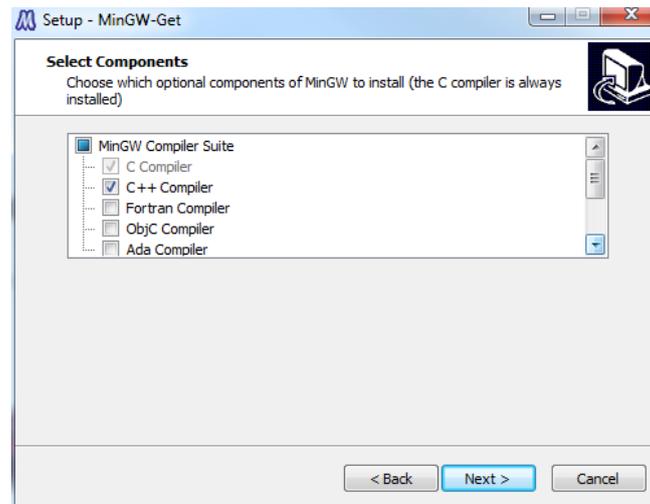


Fig. 4 Opciones de instalación MinGW

Por último es importante agregar los siguientes directorios a las variables de entorno (*PATH*):

- C:\MinGW\bin
- C:\MinGW\msys\1.0\bin

3.2. QT SDK

Desde el siguiente enlace nos podemos descargar la última versión del SDK de QT: <http://qt.nokia.com/downloads>. En el momento de hacer este tutorial, la última versión y la que es utilizada es la 1.2.1. El Qt SDK contiene las herramientas para crear aplicaciones con las librerías Qt y el entorno desarrollo Qt Creator.

Instalaremos las opciones que nos propone el instalador por defecto, aunque habrá componentes que no serán necesarios para nuestras aplicaciones.

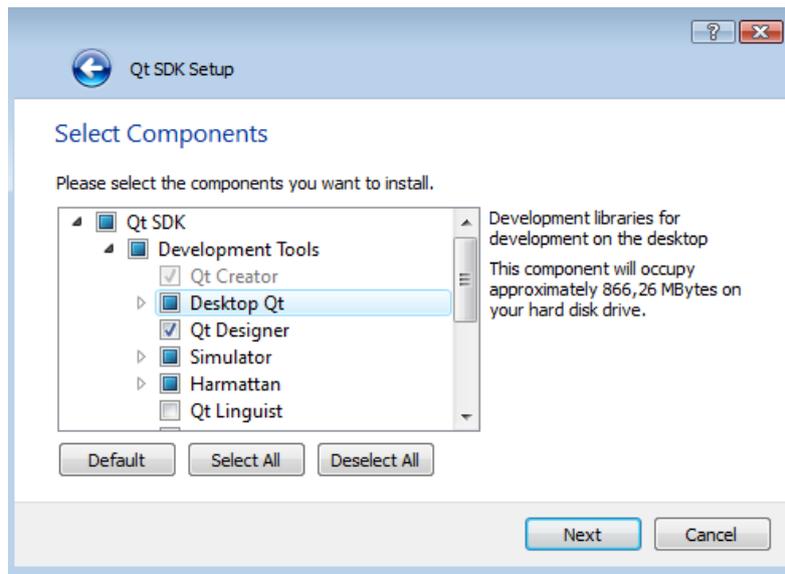


Fig. 5 Componentes QT SDK

Una vez instalado el SDK, podremos empezar a desarrollar aplicaciones con el entorno de desarrollo Qt Creator.

3.3. QT Creator

Para crear nuevos proyectos que contengan interfaces gráficas seleccionaremos:

Create new Project-> Qt4 Gui Application

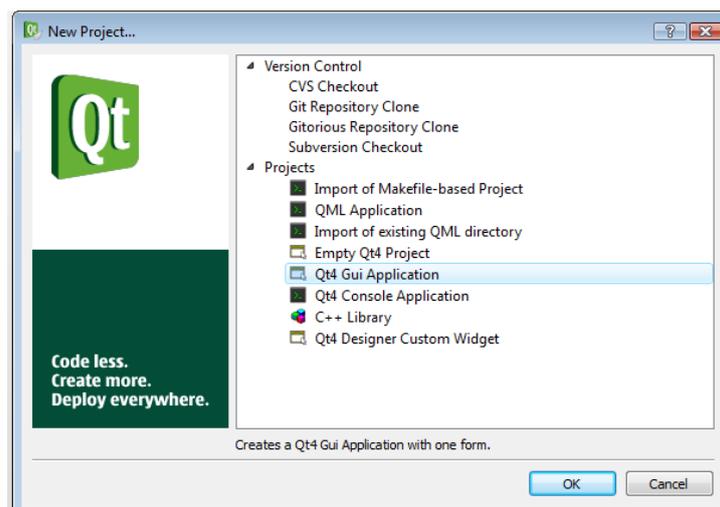


Fig. 6 Crear proyecto con Qt Creator

Después de especificar el nombre del proyecto y de los archivos principales y de la cabecera, podremos crear la interfaz gráfica de la aplicación seleccionando el archivo de la carpeta form:



Form-><Nombre del archivo>.ui

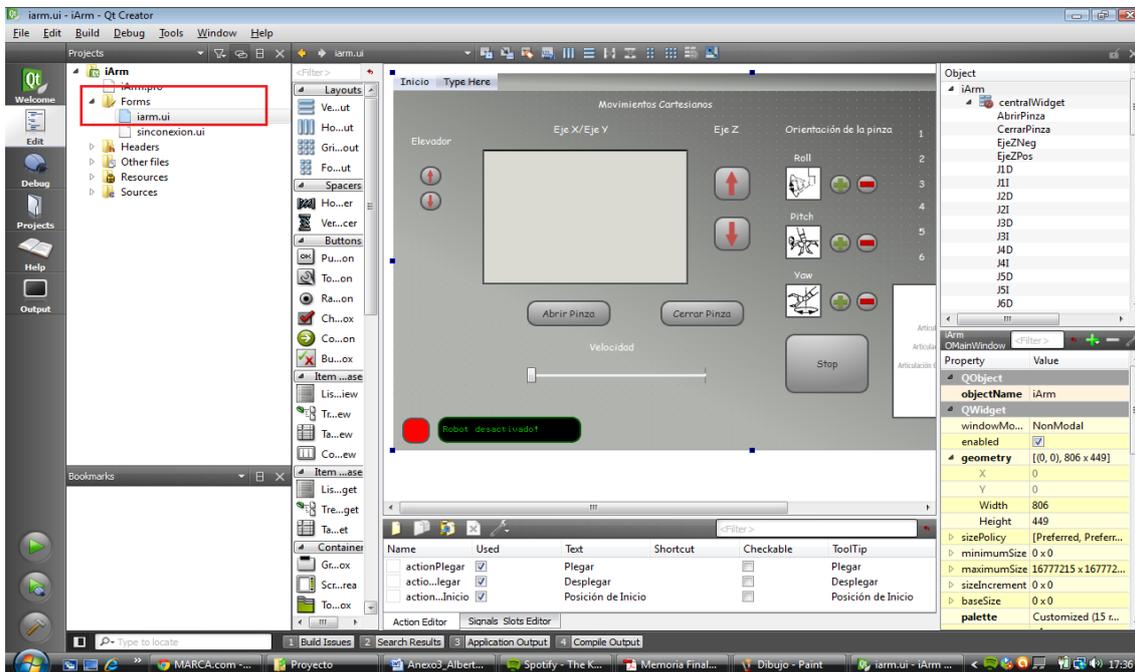


Fig. 7 Crear Interfaz gráfica con Qt Creator

4. Librería iArm

Para hacer uso de las funciones de la librería para controlar el iArm se tendrán que seguir los pasos que se explican a continuación.

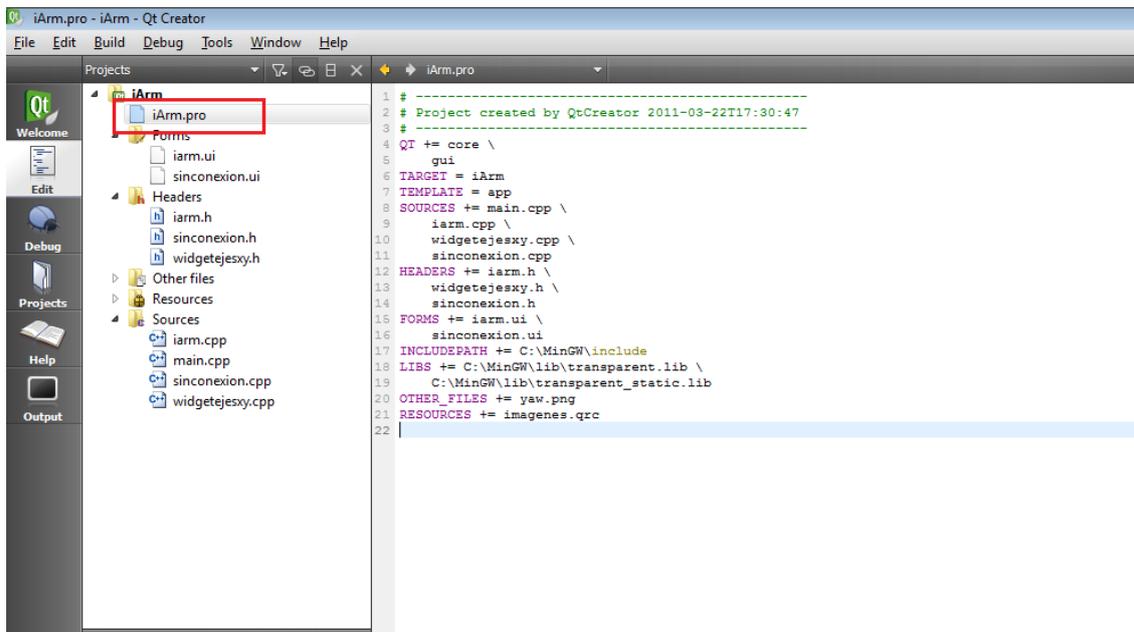
Primero tendremos que copiar en el sistema la librería Transparent.dll. En *C:\Windows\System32* (versión 32 bits) o en *C:\Windows\SysWOW64* (versión 64 bits).

Los archivos *transparent.lib* y *transparent_static.lib* los copiaremos en el siguiente directorio: *C:\MinGW\lib*

El archivo de cabecera *Tranparent.h* que define como interactuar con las librerías del iArm irá en el siguiente directorio: *C:\MinGW\include*

Una vez copiados los archivos necesarios para usar la librería del iArm, se tendrá que incluir en el archivo *<nombre del proyecto>.pro* del proyecto creado las siguientes líneas para incluir los archivos de cabecera y las librerías:

```
INCLUDEPATH += C:\MinGW\include
LIBS += C:\MinGW\lib\transparent.lib \
        C:\MinGW\lib\transparent_static.lib
```



En el código hay que incluir el archivo de cabecera Transparent.h y abrir la conexión con el iArm para empezar a usar las funciones de la librería Transparent.

```
#include <QtGui/QApplication>
#include "iarm.h"
#include "Transparent.h"
.....
#include <stdio.h>
#include <iostream>

static bool ReadArguments (int argc, char *argv[]);
void ttyreset (int signal);
void ttyset (void);

int canport = 0;
TranspHandle h = NULL;

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    iArm w;

    if(!ReadArguments(argc,argv))
        exit(0);
```



```
    ttyset();
    // Create an instance of the CAN device. You can also create your own CAN
driver interface,
    // the driver must be an implementation of the abstract CanDriver class.
    h = ArmOpen(canport);

    if(!h)
    {
        printf("Could not open CAN device on bus %d ",canport);
    }
    else{
        printf("waiting for ARM to connect on bus %d ",canport);
    }
    if(h!=NULL){
        w.setTranspHandle(h);
    }
    w.show();

    return a.exec();
}
static bool ReadArguments(int argc, char *argv[])
{
    for (int i = 1; i < argc; i++)
    {
        char *ptr = argv[i];
        while (*ptr == '-')
            ptr++;
        const char prm = *(ptr++);
        const bool arg = (*ptr == '=');
        if(arg) ptr++;

        switch(prm)
        {
            case 'p':
                if(arg)
                    canport = strtoul(ptr, NULL, 10);
                else
                {
                    std::cout << "please supply a port number with -p (e.g. '-p=1')" <<
std::endl;
                    return false;
                }
                break;
            case '?':
```



```
        default:
            std::cout << "\"\" << argv[0] << "\": unknown commandline argument:
" << prm << std::endl;
            return false;
        }
    }
    return true;
}

void ttyreset(int signal)
{
#ifdef UNIX
    tcsetattr(STDIN_FILENO, TCSANOW, &ots);
#endif
    exit(signal);
}

void ttyset(void)
{
#ifdef UNIX
...
#endif
}
```

5. Librería SDL

Para usar el *joystick* se ha utilizado la librería SDL. En este manual se ha utilizado la versión 1.3, que permite acceder a las funcionalidades del *joystick* y si dispone de *forcefeedback* hacer uso de él. Desde el siguiente enlace se puede descargar la última versión: <http://www.libsdl.org/hg.php>.

El archivo comprimido con la última versión de la biblioteca contendrá dos directorios importantes:

- include
- bin

Todo lo que se encuentre en include hay que colocarlo en un directorio nuevo: C:\MinGW\include\SDL\. Para los archivos en lib con extensión .a y .lib deben ir en: C:\MinGW\lib\. Los archivos en lib con extensión .dll van en C:\Windows\System32\ (versión 32 bits) o en C:\Windows\SysWOW64



(versión 64 bits). En el caso del paquete principal de SDL, el archivo SDL.dll, se encuentra en el directorio bin y habrá que ponerlo también en en C:\Windows\System32\.

Una vez copiados los archivos necesarios para usar la librería SDL, se tendrá que incluir en el archivo *<nombre del proyecto>.pro* del proyecto creado las siguientes líneas para incluir los archivos de cabecera y las librerías:

```
INCLUDEPATH += C:\MinGW\include\SDL\  
LIBS += C:\MinGW\lib\libSDLmain.a \  
C:\MinGW\lib\SDL.dll \
```

En el código hay que incluir las referencias a los archivos de cabecera:

```
#include <SDL.h>  
#include "SDL_haptic.h"
```

Mediante las siguientes líneas en el código se iniciará el SDL para usar el *joystick* y su *forcefeedback*.

```
//Initialize SDL Joystic  
if (SDL_Init(SDL_INIT_JOYSTICK | SDL_INIT_HAPTIC) < 0) {  
    printf("Unable to init SDL: %s\n", SDL_GetError());  
}  
joystick = SDL_JoystickOpen(0);
```