

ARQUITECTURA DE CONTROL MULTIMODAL PARA ROBÓTICA ASISTENCIAL

José M. Catalán, Jorge A. Díez, Arturo Bertomeu-Motos,
Francisco J. Badesa, Rafael Puerto, José M Sabater, Nicolas Garcia-Aracil

Neuroingeniería Biomédica, Universidad Miguel Hernández de Elche
{jose.catalan, jdiez, abertomeu, fbadesa, r.puerto, j.sabater, nicolas.garcia}@umh.es

Resumen

Este documento presenta una arquitectura de control multimodal para robótica de asistencia, la cual trata de tener en cuenta las decisiones del usuario para mejorar en el desempeño de las tareas al mismo tiempo que se implementa un método para minimizar posibles errores en el manejo del robot mediante un control visual. A través de la información proporcionada por el sistema de eye-tracking, el usuario será capaz de interactuar con el sistema para seleccionar el objeto deseado, indicar la intención de cogerlo o incluso abortar la ejecución. El sistema incorpora un sistema de tracking 3D para conocer la ubicación de los objetos con respecto al manipulador robótico. Este sistema nos sirve tanto para definir la posición que debe alcanzar el robot, como para corregir las posibles desviaciones durante la ejecución de la trayectoria.

Palabras clave: Interface multimodal, Robótica asistencial, ROS

1 INTRODUCCIÓN

Hoy en día, las tecnologías de asistencia desarrolladas para apoyar en el desempeño de actividades de la vida diaria, movilidad, comunicación y demás, tienden a basarse en la integración de las capacidades del usuario junto con la tecnología [3] [2]. La mejora de esta integración y cooperación puede dividirse en tres áreas principales: 1) mejoras en los dispositivos de asistencia, refiriéndonos a las partes mecánicas, electrónicas, etc.; 2) mejoras en el interfaz con el que interactúa el usuario con el sistema; y 3) mejoras en el control compartido entre el usuario y la tecnología asistencial. Sin embargo, a pesar del rápido progreso en el área de la robótica, todavía no se han alcanzado los requisitos necesarios para interactuar cuidadosamente con personas, objetos u obstáculos en un entorno real [5][1][4]. Por ende, aun no existe un sistema completo que permita a los usuarios con discapacidad llevar a cabo adecuadamente actividades de la vida diaria en un entorno altamente cambiante. El objetivo principal de este trabajo

es presentar una posible solución al planteamiento de una arquitectura de control centrada en la interacción con el usuario y la minimización de errores en el control del robot.

2 PROTOCOLO

Primeramente se definió un protocolo de actuación el cual define que actividad se va a realizar y que proceso de ejecución se va a llevar a cabo. La actividad desarrollada para la experimentación consistirá en beber. A modo de resumen, en la Figura 1 se muestra un esquema de los pasos principales definidos en este protocolo. Se han separado en bloques de colores los procesos principales: 1) El usuario selecciona el vaso a coger mediante el sistema de eye-tracking. El proceso consistirá en mirar fijamente un objeto durante un periodo determinado de tiempo; 2) Se recoge la posición del vaso y se ejecuta el control visual para conducir al robot hasta la posición de pre-grasp; 3) En el momento en que el usuario lo indique, se

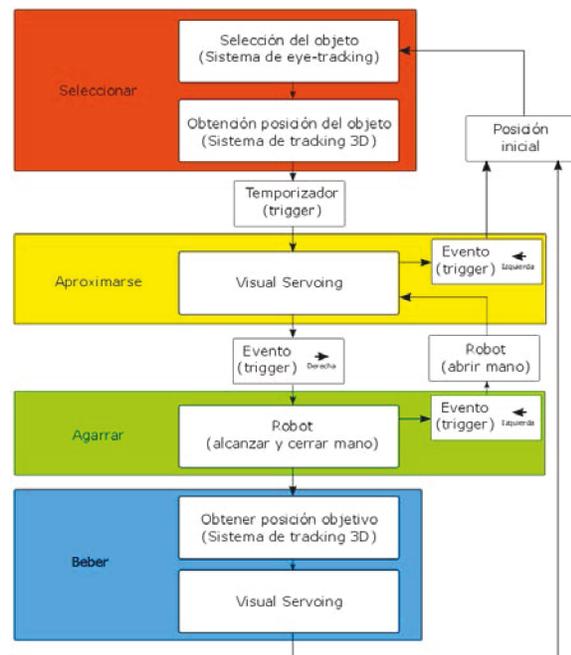


Figura 1: Definición del proceso de ejecución de la actividad

procederá a coger el objeto; 4) Una vez hayamos cogido el objeto lo llevaremos donde proceda. Puede apreciarse que en el caso de los bloques de aproximación y agarre se implementa un evento que sirve para abortar la tarea actual y volver al estado anterior.

2.1 SISTEMA DE EYE-TRACKING

Se han utilizado las Tobii Pro Glasses 2. Este dispositivo inalámbrico de eye-tracking envía simultáneamente datos y Video (full HD, 1920x1080, 25 fps, comprimido en H.264) en tiempo real mediante comunicación WiFi. Dentro de todo el conjunto de datos se encuentra la posición de la mirada definida en función de la imagen capturada por la cámara frontal de las gafas. Primeramente, definimos un tamaño de ROI (Región de interés) en función del tamaño del objeto, la distancia y la estabilidad de la mirada. De esta manera, como observamos en el Figura 2, centrando esta ROI en la posición de la mirada proporcionada por las gafas, definiremos una región donde consideraremos que el usuario está mirando. Esta ROI nos permite delimitar un trozo de imagen al cual aplicaremos un algoritmo de detección de colores basado en el modelo de color HSV. El resultado de este filtro puede observarse en las tres pequeñas ventanas que aparecen a la derecha de la Figura 2, cada una correspondiente a uno de los tres colores definidos para los vasos. Esto nos permitirá discernir entre los distintos objetos, a la vez que reducimos la carga computacional por el echo de procesar únicamente un trozo de la imagen. Para detectar el objeto, calculamos el porcentaje correspondiente a cada uno de los colores que buscamos. En el momento en que este porcentaje su-

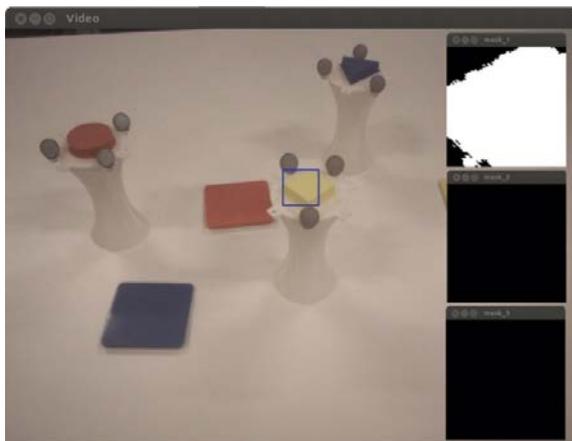


Figura 2: Captura del funcionamiento de las Tobii Pro Glasses 2, donde se observa la ROI definida sobre la imagen capturada por la cámara, junto con tres ventanas independientes en representación del resultado obtenido de los tres filtros de color

per cierto umbral durante un periodo de tiempo definido consideraremos que el objeto a sido seleccionado.

2.1.1 Detección de eventos

Empleando la posición de la mirada proporcionada por las Tobii Pro Glasses 2, podemos definir eventos que funcionarían con la misma lógica que lo harían los comandos EOG. Como se observa en la Figura 3, controlando los ejes correspondientes a la posición de la mirada, podemos definir dos señales que nos permiten detectar cuando miramos en una u otra dirección.

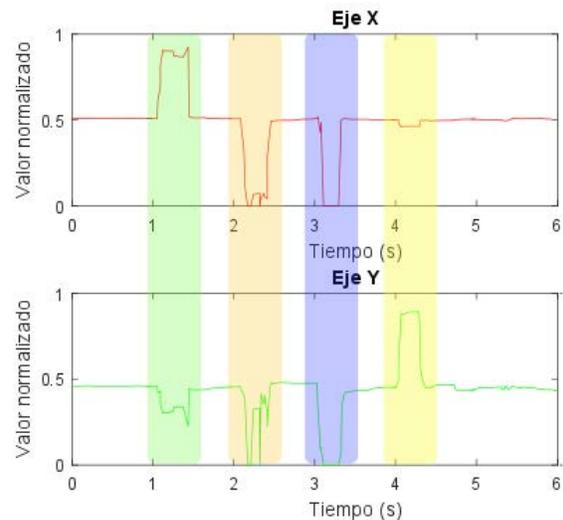


Figura 3: Señales en bruto correspondientes a los valores de las coordenadas X e Y de la posición de la mirada proporcionada por las Tobii Pro Glasses 2, donde se indican los comandos detectados: derecha (verde), izquierda (rojo), arriba (azul) y abajo (amarillo)

2.2 SISTEMA DE TRACKING 3D

La cámara OptiTrack Trio emplea pequeñas esferas recubiertas de un material reflector de infrarrojos (trackers) para calcular la posición y orientación de los trackers en 3D. Para ello, es necesario emplear el software oficial de Optitrack llamado MOTIVE, el cual permite definir grupos fijos de como mínimo 3 trackers, de los cuales nos dará posición y orientación. Este software solamente tiene compatibilidad con Windows. Esto nos obliga a enviar la información a ROS por medio de VRPN (Virtual-Reality Peripheral Network), que es el modo en el que MOTIVE permite extraer los datos. Este sistema proporciona en tiempo real la posición y orientación de los objetos que se encuentran dentro del campo de visión de la cámara y que obviamente hayan sido definidos previamente en MOTIVE. Para evaluar

la robustez del sistema Optirack se han realizado una serie medidas. Los resultados pueden observarse en la Figura 4. En la primera gráfica observamos el movimiento descrito por el objeto. Entre dos de sus tracker se ha definido una longitud fija la cual vamos a intentar calcular. El resultado de esta calculo se encuentra representado en la segunda gráfica. Mediante esta información hemos extraído la distribución normal de los datos de la cámara. Hemos obtenido una media de 10.296cm , para una longitud real de 10.3cm . La desviación estándar es de 0.022 . Y el máximo error cometido es de 7.8mm .

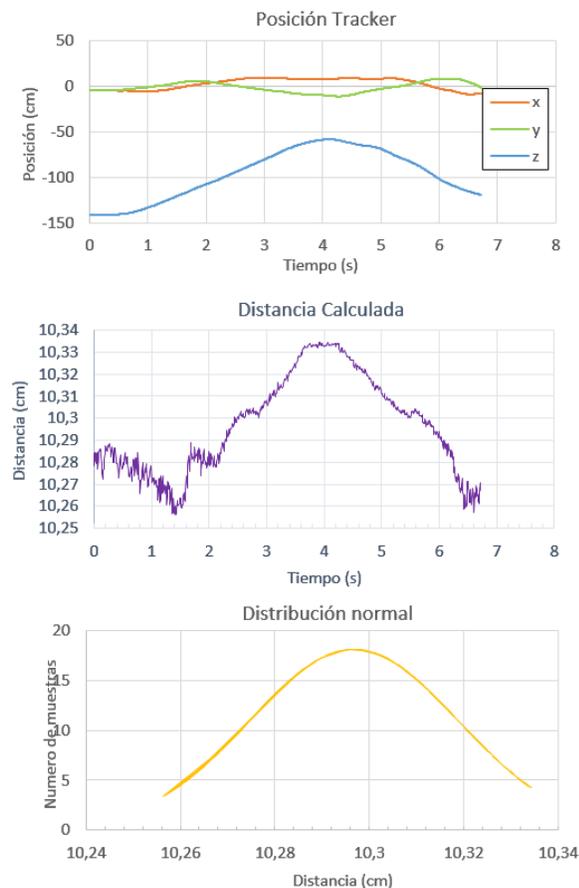


Figura 4: Evaluación de la información proporcionada por la cámara de tracking 3D Optitrack Trio

2.3 ROBOT MANIPULADOR

El robot manipulador empleado es el Brazo Kinova JACO² con la pinza de tres dedos. Su control esta basado en el paquete de ROS desarrollado por WPI (Worcester Polytechnic Institute), aunque se han realizado muchas modificaciones para adaptarlo a nuestros requerimientos. Por ejemplo, se ha desarrollado un nuevo nodo llamado *nbio_jaco_controller*, el cual puede observarse en la Figura 6. Este actúa como middleware entre el

controlador de WPI (*wpi_jaco_wrapper*) y nuestras aplicaciones, con la intención de implementar más funcionalidades, facilitando el futuro control de robot por nuestra parte.

2.4 CONTROL VISUAL

En aplicaciones como esta es necesario considerar un error variable en el control del robot debido a imperfecciones en el proceso de calibración, base móvil, o cualquier otra fuente que impida mover el robot con la precisión suficiente como para realizar la aproximación al objeto de forma adecuada. El control planteado en este documento consiste en minimizar el error entre el objetivo y el efector final. Para ellos realizamos el siguiente planteamiento. En distancias cortas puede observarse que el error acumulado en el control cartesiano del efector final normalmente es pequeño. En este caso, podemos asumir que el error tiene una dependencia lineal con el movimiento. Esta consideración permite segmentar la trayectoria en pequeños tramos, y tras la ejecución de cada uno de ellos, comprobar el error cometido y compensarlo en la ejecución del siguiente. Además, esto permite alcanzar objetos en movimiento, o lo que es lo mismo, alcanzar objetos mientras la base móvil del robot se mueve, ya que se realizan correcciones en tiempo real durante la ejecución de la trayectoria. En la Figura 5 se resume el diagrama de estados que implementa el control visual.

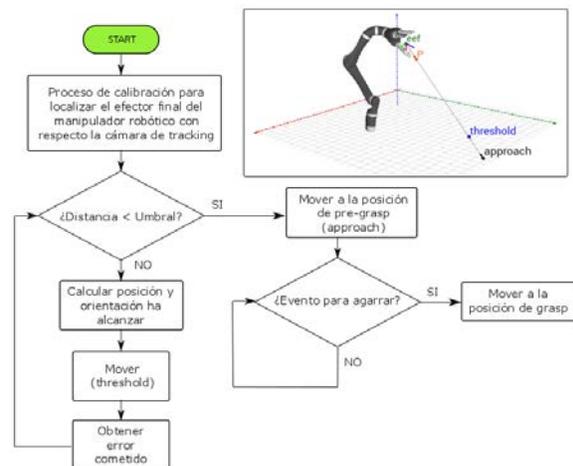


Figura 5: Resumen de la maquina de estados finitos para la ejecución del control visual

3 ARQUITECTURA DE ROS

En la Figura 6, podemos ver un resumen de la interconexión de los nodos principales implementados en ROS. Por un lado, tenemos el paquete *wpi_jaco_wrapper* implementado por WPI, el cual

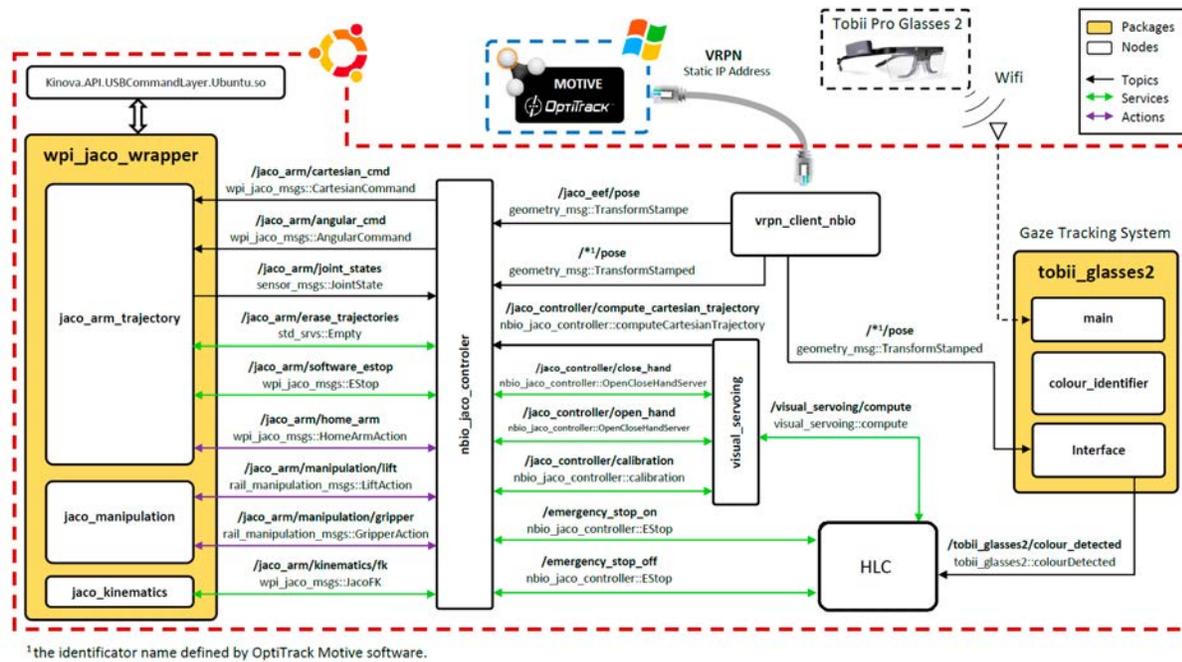


Figura 6: Arquitectura implementada en ROS. Resumen de las comunicaciones entre los nodos principales

corresponde a nuestro controlador de más bajo nivel. Por encima de este tenemos nuestro nodo *nbio_jaco_controller*. Este nodo aprovecha las funcionalidades del *wpi_jaco_wrapper* para implementar nuevas herramientas que nos permiten tener un mayor control y conocimiento del estado del robot. Además, integra a JACO² con el sistema de tracking, de manera que nos facilita la calibración del robot con respecto a la cámara. Para ello se emplea la información que aporta el nodo *vrpn_client_nbio*, el cual se encarga de la comunicación entre MOTIVE y ROS, recogiendo todos los datos de los trackers definidos en este software. El control visual ha sido implementado en una máquina de estados finitos definida en el nodo *visual_servoing*. Este nodo nos permite definir un objetivo, y en base a este, el diagrama ejecutará el proceso de la Figura 5.

Por otro lado tenemos el paquete *tobii_glasses2*, que es el encargado de trabajar con las Tobii Pro Glasses 2. En este paquete se implementa la lectura de vídeo y datos además de realizar el proceso de selección del vaso objetivo.

4 PRUEBAS EXPERIMENTALES

Para comprobar el rendimiento del sistema en el desempeño de la actividad se ha llevado a cabo la experimentación expuesta en la Figura 7. El experimento consiste en lo siguiente: 0) Mediante el uso del sistema de eye tracking el usuario selecciona

el vaso que se desea agarrar; 1) En el momento en que se ejecuta la selección del objeto, el sistema obtiene la posición del objeto seleccionado por el usuario mediante la cámara de tracking y ejecuta el control visual para mover el robot hasta la posición de pre-grasp; 2) Cuando la posición de pre-grasp es alcanzada, el sistema sitúa al robot en la posición de grasp y espera a que el usuario indique que el movimiento ha sido correcto y desea coger dicho vaso mediante una mirada a la derecha (evento); 3) En este caso, el usuario todavía no ha indicado que deba cogerse el vaso, pero este se ha sido desplazado intencionadamente. Por tanto, vemos como el sistema corrige la posición del robot para poder coger el vaso; 4) En este momento el usuario ha indicado que se desea coger el vaso; 5) Y por último, se ejecuta el proceso para acercar el vaso al usuario para en este caso, permitirle beber.

En la gráfica que se observa en la Figura 7, se representa la posición del efector final con respecto a la posición del vaso para dos casos: con el vaso fijo (paso 1 y 2) y segundo con el vaso en movimiento (paso 3). Como puede observarse, el efector final alcanza la posición objetivo adecuadamente. Cabe destacar que la coordenada Z presenta un offset con respecto al tracker del vaso definido para permitir el agarre del vaso.

5 CONCLUSIONES

En este trabajo se ha presentado las bases de una arquitectura de control multimodal para

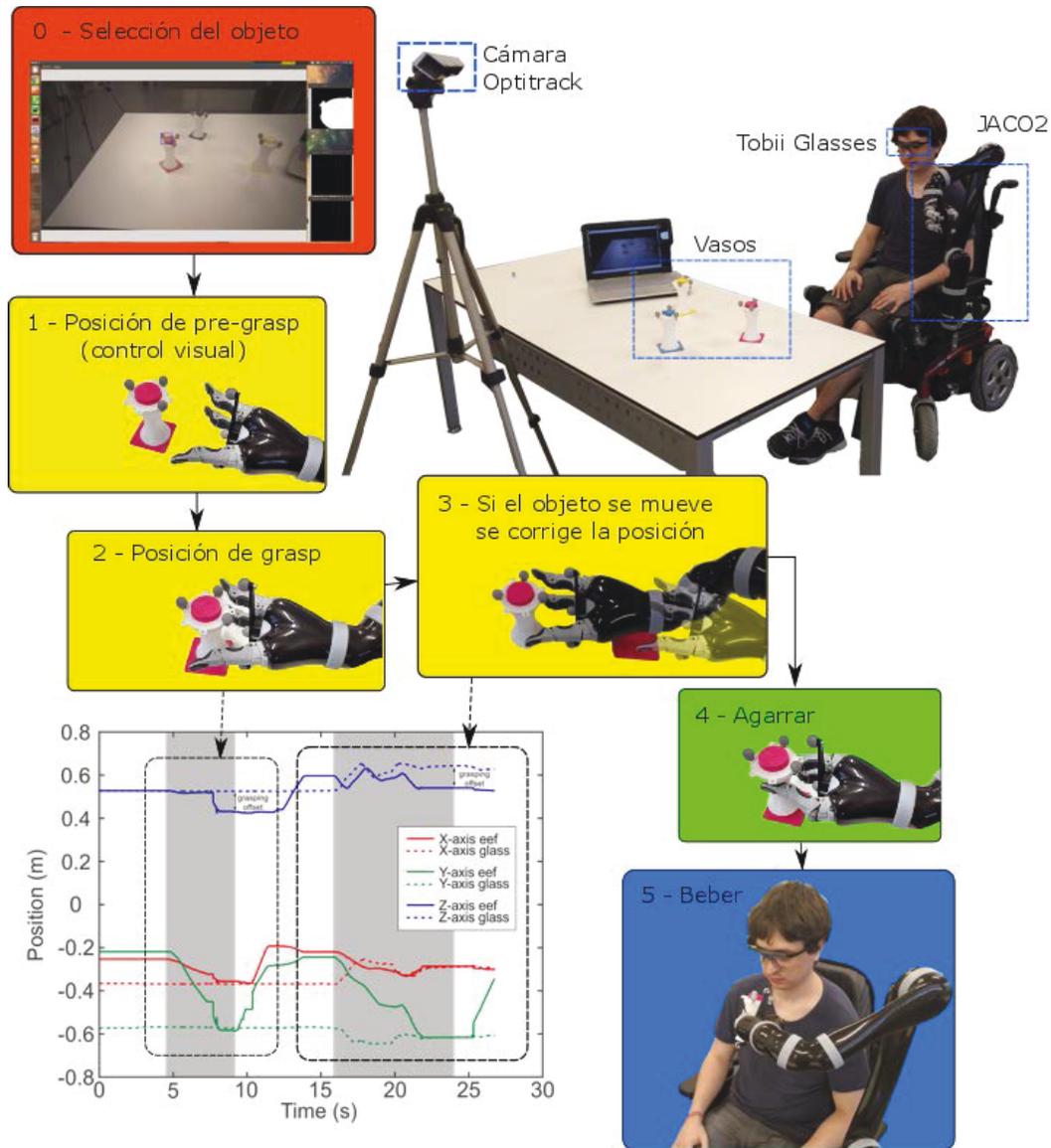


Figura 7: Representación gráfica de la experimentación llevada a cabo

robótica asistencial, incorporando tecnología eye-tracking, proporcionada por las Tobii Pro Glasses 2, y tracking 3D mediante la cámara Optitrack Trio para controlar el robot manipulador JACO². Como se ha demostrado, el sistema consigue desempeñar la actividad propuesta permitiendo al usuario modificar el proceso de ejecución de la arquitectura de control. Se va a continuar trabajando en esta línea para intentar mejorar la adaptación del sistema con el usuario, a la vez que intentar incorporar más inteligencia al sistema para que sea capaz de evitar otras posibles fuentes de error propias de un ambiente cambiante como es el entorno real.

Agradecimientos

Este trabajo ha sido financiado por la Comisión Europea a través del proyecto AIDE: Adaptive

Multimodal Interfaces to Assist Disabled People in Daily Activities (Grant agreementno: 645322) y por el Ministerio de Economía y Competitividad a través del proyecto DPI2015-70415-C2-2-R.

Referencias

- [1] A. Bicchi, M. Bavaro, G. Boccadamo, D. De Carli, R. Filippini, G. Grioli, M. Piccigallo, A. Rosi, R. Schiavi, Soumen Sen, G. Tonietti, (2008) *Physical human robot interaction: dependability, safety, and performance*, International Workshop on Advanced Motion Control, pp. 9-14.
- [2] J. Vogel, S. Haddadin, J. D. Simeral, S. D. Stavisky, D. Bacher, L. R. Hochberg, J. P. Donoghue, P. van der Smagt, (2010)

Continuous Control of the DLR Light-weight Robot III by a human with tetraplegia using the BrainGate2 Neural Interface System, In International Symposium on Experimental Robotics.

- [3] R. E. Cowan, B. J. Fregly, M. L. Boninger, L. C. Chan, M. M. Rodgers, D. J. Reinkensmeyer, (2012) *Recent trends in assistive technology for mobility*, J of NeuroEngineering and Rehab, pp. 9-20.
- [4] R. Ying, J. Weisz, P. K. Allen, (2015) *Grasping with your brain: a brain-computer interface for fast grasp selection*, Workshop on Robotic Hands, Grasping, and Manipulation.
- [5] S. Haddadin, A. Albu-Schaffer, G. zinger, (2009) *Requirements for Safe Robots: Measurements, Analysis and New Insights*, International Journal on Robotics Research, pp. 1507-1520.