

HERRAMIENTA PARA VISUALIZACIÓN GRÁFICA DE FUERZAS DE CONTACTO Y DE MOVIMIENTOS DE UNA MANO ROBÓTICA CON SENSORES TÁCTILES

Andrés Montaña – Raúl Suárez
Universitat Politècnica de Catalunya (UPC), Barcelona.
andres.felipe.montano@upc.edu – raul.suarez@upc.edu

Resumen

En este trabajo se presenta una herramienta de software que permite la visualización gráfica de fuerzas de contacto de especial utilidad en prensiones usando manos robóticas. Las fuerzas visualizadas se generan debido al contacto entre las yemas de los dedos y el objeto manipulado, o debido a los pares aplicados en las articulaciones de los dedos. La herramienta también permite mover la mano real, ya sea especificando configuraciones de la mano o mediante instrucciones individuales para cada una de las articulaciones, el movimiento es ejecutado al mismo tiempo que se visualiza en la simulación gráfica. Como ejemplo práctico, se presenta una aplicación usando la mano Allegro con sensores táctiles WTS-FT y se muestran diferentes ejemplos de prensión de objetos con la información táctil registrada en los sensores táctiles y las fuerzas calculadas usando los pares aplicados por los motores.

Palabras clave: Sujeción, Manipulación, Sensores Táctiles, Robótica.

1 INTRODUCCIÓN

Cuando se desarrollan componentes de software para aplicaciones en robótica se deben tener en cuenta varios aspectos para obtener una herramienta completa. Usualmente se recomienda el uso de C++ como lenguaje de programación ya que este es un lenguaje de alto nivel que cuenta con técnicas eficientes de compilación para optimizar las aplicaciones, además permite el uso de técnicas avanzadas de programación como punteros inteligentes y manejo de excepciones, las cuales son necesarias para desarrollar aplicaciones robustas. También es deseable que los componentes desarrollados sean multiplataforma y que las dependencias de los mismos sean fáciles de configurar. En un entorno científico se busca que el software sea de libre acceso, ya que de esta forma es posible reutilizar y compartir los resultados y las soluciones con el resto de la comunidad.

En la actualidad existen múltiples herramientas de software asociadas al mundo de la robótica que difieren entre sí en las funcionalidades, tipos de licencias y complejidad que cada una ofrece. Por ejemplo, para simulación de robots, considerando propiedades físicas del robot y el entorno, hay librerías como Open Dynamics Engine (ODE)[9] o Bullet[1]; para realizar la planificación de movimientos hay librerías como Open Motion Planning Library (OMPL) [10], Task-Motion Kit [2], o Kautham project [8]; también hay herramientas para simulación y obtención de prensiones como GraspIt [5], OpenGrasp [4] o Simox [11].

En este trabajo se presenta una herramienta que permite la visualización gráfica de una mano robótica, de las fuerzas que esta está ejerciendo en las yemas de los dedos debido a los pares aplicados en sus articulaciones, y las realmente registradas en caso de disponer de sensores táctiles. Asimismo, la herramienta desarrollada también permite mover la mano real sincronizadamente con la simulación gráfica, ya sea especificando la configuración completa de la mano o mediante instrucciones individuales para cada una de sus articulaciones.

El resto del artículo está organizado de la siguiente forma. En la Sección 2 se presenta una visión general de la estructura de software en la que se basa la herramienta propuesta, así como una descripción de las librerías que se utilizan. En la Sección 3 se describe un ejemplo usando la mano Allegro y los sensores táctiles WTS-FT. Los resultados experimentales de la prensión de diferentes objetos se presentan en la Sección 4. Finalmente, las conclusiones y la propuesta de trabajo futuro se presentan en la Sección 5.

2 DESCRIPCIÓN GENERAL DE LA HERRAMIENTA DESARROLLADA

La herramienta que se presenta está implementada por completo en C++, y utiliza librerías estándar que permiten hacer más fácil el mantenimiento y desarrollo de la misma. En la Figura 1 se muestra

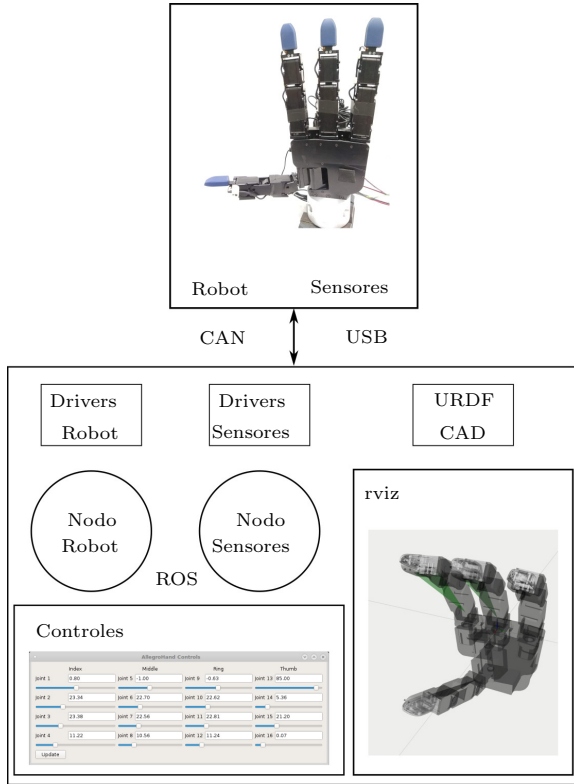


Figura 1: Diagrama de bloques con los componentes (hardware y software) de la herramienta presentada.

un diagrama de bloques con los componentes hardware y software forman el sistema. El hardware incluye la mano robótica, los sensores y los buses de comunicaciones para conectarlos al sistema de cómputo. El software está compuesto por los diferentes módulos que conforman la herramienta y que se detallan más adelante.

Dentro de las principales librerías que se usan en la herramienta se encuentra *Robot Operating System* (ROS)[7] que funciona como una capa de comunicaciones que permite interconectar de forma simple los diferentes módulos de software que componen la herramienta. ROS está basado en un arquitectura de tipo grafo en la que los procesos, llamados nodos, se organizan en una red *peer-to-peer* y se acoplan de forma flexible a través de una infraestructura de comunicaciones, es decir, en los nodos se realiza el procesamiento de la información y las rutinas para enviar y recibir mensajes. El intercambio de mensajes entre nodos se realiza bajo dos esquemas: servicios (*services*) y temas (*topics*). Los servicios tienen una interacción del tipo petición–respuesta, mientras que los temas utilizan publicadores (*publishers*) y subscriptores (*subscribers*) para intercambiar información.

Por otro lado, para describir la mano robótica

se utiliza el formato unificado de descripción de robots (*URDF*) basado en un formato XML que permite representar no solo modelos de robots, sino también modelos de sensores o incluso escenas. Para visualizar el robot dentro de la herramienta se utiliza el modelo CAD que contiene las mallas triangulares de cada uno de los segmentos (*links*) que conforman el robot, así como el modelo de los sensores.

La visualización de la mano robótica se realiza mediante una escena de *rviz* [3], que es una herramienta de visualización en 3D para aplicaciones de ROS. *rviz* no solo proporciona una vista del modelo de la mano, sino que también permite visualizar información de los sensores. Las fuerzas medidas por los sensores se representan mediante marcadores (*markers*) que contienen la información de la ubicación del punto de contacto y la magnitud de la fuerza. La fuerza aplicada por cada uno de los dedos sobre el objeto manipulado también se puede calcular utilizando la información de los pares aplicados a cada una de las articulaciones del dedo y la matriz Jacobiana para una configuración dada, usando la siguiente relación:

$$F = J(q)^{T*} \tau$$

donde $J(q)^{T*}$ es la inversa de la transpuesta de la matriz Jacobiana, que depende de los valores articulares, y τ es el vector de los pares aplicados para mantener una configuración del dedo. La fuerza que ejercen las articulaciones de los dedos sirve para compensar el peso de sus falanges y producir las fuerzas de prensión. Este efecto se tiene en cuenta en el cálculo de la fuerzas aplicadas en los extremos de los dedos a partir de los torques aplicados en las articulaciones.

Para mover la mano robótica real, se desarrolló una interfaz gráfica basada en QT [6], que es un entorno de trabajo multiplataforma orientado a objetos para desarrollar programas que utilicen una interfaz gráfica. Esta interfaz gráfica cuenta con controles individuales para cambiar la configuración de cada una de las articulaciones del robot descrito mediante el archivo URDF.

3 APLICACIÓN USANDO LA MANO ALLEGRO

A continuación se presenta un ejemplo de uso de la herramienta en el que se utiliza una versión de mano izquierda de la mano robótica Allegro de Wonik Robotics. Esta mano tiene un diseño antropomorfo con cuatro dedos, cada uno con cuatro grados de libertad (*GDL*), como se muestra en la Figura 2. Los dedos índice, medio y anular tiene la misma estructura cinemática. El dedo

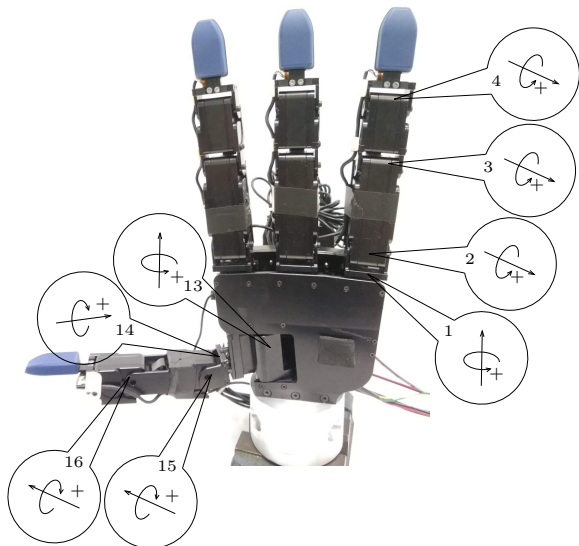


Figura 2: Mano antropomorfa Allegro, en su versión de mano izquierda, con sensores táctile (WTS-FT).

pulgar es diferente de los otros tres y tiene un *GDL* que le permite realizar el movimiento de abducción para sujetar objetos.

La mano Allegro en su versión comercial no tiene sensores táctiles, por este motivo las yemas de los dedos originales han sido reemplazadas por yemas de dedos con los sensores WTS-FT de Weiss Robotics. Cada sensor está compuesto por una matriz táctil resistiva de 4×8 taxels cuadrados de 3,8 mm de lado, cada taxel retorna un valor entre 0, cuando no se aplica presión, y 4095, cuando se aplica la máxima presión medible. Generalmente el contacto entre el objeto y cada una de las yemas de los dedos se produce en una región de contacto, en este trabajo se considera el baricentro de dicha región como el punto de contacto efectivo entre el objeto y la yema del dedo. Para incluir la información de la localización del punto de contacto se utilizan dos articulaciones prismáticas virtuales, $d_1(x)$ y $d_2(y)$, que añaden dos *GDL* no actuados adicionales al dedo que se ubican al final de la cadena cinemática real. Asumiendo que la superficie del sensor es plana, d_1 y d_2 permiten ubicar el punto de contacto en cualquier posición del plano. La Figura 3 muestra el sensor WTS-FT y la distribución de los taxels que lo forman. También se muestra un ejemplo de una región de contacto remarcada con una elipse, donde se pueden apreciar los diferentes valores registrados en los taxels que se representan mediante una escala de colores, además se indica el baricentro (considerado como el punto de contacto efectivo entre el objeto y la yema del dedo) y las articulaciones virtuales que desplazan el punto de

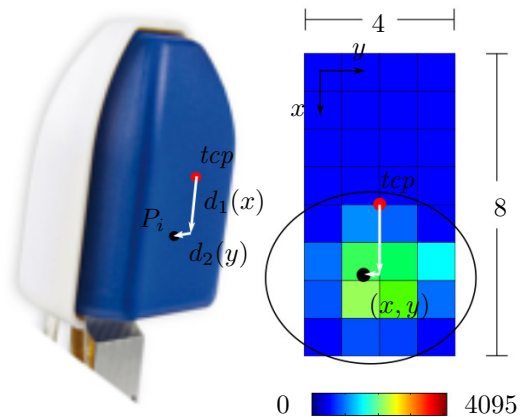


Figura 3: Sensor WTS-FT de Weiss Robotics y una representación de los taxels de que dispone, la barra en la parte inferior relaciona los colores representados en los taxels con las medidas registradas por el sensor.

contacto desde el extremo del dedo (*tcp*) hasta el punto de contacto correspondiente en la superficie del sensor.

En la Figura 4 se muestra una captura de pantalla de la herramienta presentada cuando se ejecuta usando el modelo de la mano Allegro con los sensores táctiles WTS-FT. En la captura de pantalla de la interfaz gráfica se muestran cinco ventanas, cada una identificada con un número dentro de un círculo en la imagen. La primera ventana corresponde a los nodos que conectan los controladores (*drivers*) de la mano y los sensores, los cuales establecen la comunicaciones a nivel físico a través de los puertos CAN y USB, para la mano y los sensores, respectivamente. La segunda ventana corresponde a la interfaz gráfica basada en *rviz* para la visualización de los modelos de la mano y de los sensores, y en la que se visualizan las fuerzas registradas por los sensores y las calculadas mediante los pares aplicados a las articulaciones de la mano. La tercera ventana corresponde al nodo de ROS que se comunica la interfaz gráfica con los controles individuales de cada una de las articulaciones y el nodo ROS con el controlador de la mano. La cuarta ventana corresponde a la interfaz gráfica para controlar el movimiento individual de las articulaciones de la mano. La quinta ventana corresponde al nodo ROS que permite la visualización de las fuerzas registradas por los sensores y además calcula las fuerzas ejercidas por los dedos a partir de los pares aplicados en cada articulación.

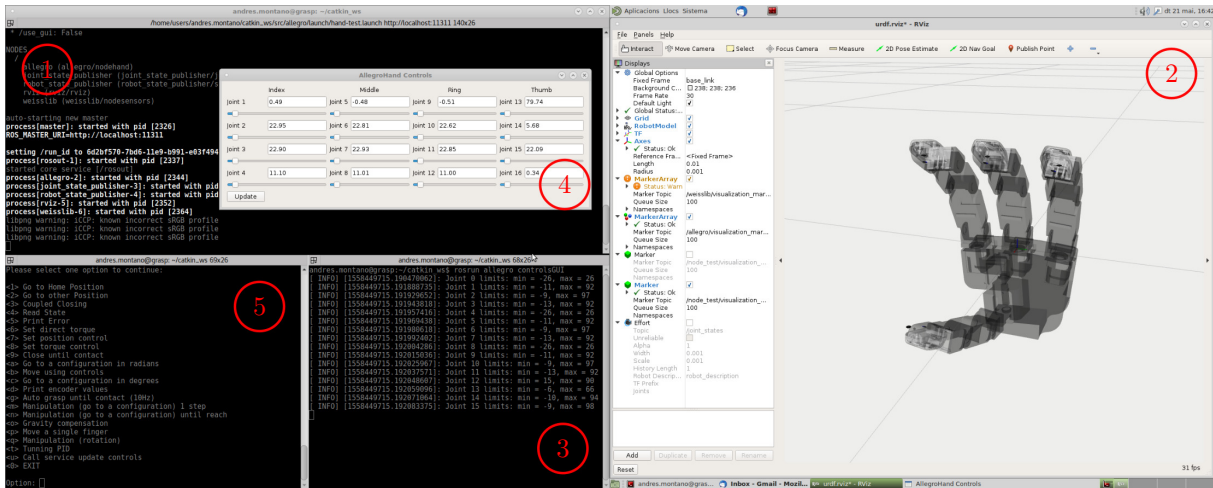


Figura 4: Captura de pantalla de la herramienta cuando se ejecuta un ejemplo usando la mano Allegro equipada con sensores táctiles.

4 RESULTADOS EXPERIMENTALES

A continuación se presentan diferentes ejemplos en los que se muestran las principales funcionalidades de la herramienta presentada. En la Figura 5 se muestran tres ejemplos en los que se interactúa con la mano robótica y los sensores táctiles para generar fuerzas sin sujetar un objeto y que son visualizadas en la interfaz gráfica. En los tres ejemplos la mano se colocó en una configuración predeterminada y se mantuvo estática en dicha posición. Las fuerzas registradas por el sensor táctil se representan con flechas de color azul, mientras que las calculadas con los pares aplicados en las articulaciones se representan con flechas de color rojo. En el primer ejemplo se ejerce presión directamente sobre uno de los sensores táctiles y en la interfaz gráfica se visualiza la fuerza registrada por el sensor. En el segundo ejemplo se empuja el dedo desde la parte posterior de la yema del dedo intentando cambiar la configuración del mismo. Como resultado de esta acción el controlador de la mano incrementa el par en las articulaciones del dedo para mantener la configuración predeterminada, el incremento de los pares produce una fuerza que se visualiza en la interfaz gráfica. En el tercer ejemplo se realiza una combinación de las acciones de los dos ejemplos anteriores, empujando el dedo desde la yema intentando cambiar la configuración del mismo. En este caso se visualizan simultáneamente la fuerza registrada por el sensor y la fuerza calculada debido a los pares aplicados en las articulaciones.

En la Figura 6 se presentan cuatro ejemplos de presión de diferentes objetos. En cada ejemplo se

muestra una imagen con la presión real ejecutada por la mano y dos capturas de pantalla de la interfaz gráfica donde se muestra, en la primera captura, las fuerzas calculadas usando los pares aplicados en las articulaciones, y en la segunda captura, las fuerzas registradas en los sensores táctiles. En el primer ejemplo se sujeta un objeto de forma esférica usando dos dedos; en el segundo ejemplo se sujeta una botella con caras planas usando tres dedos; en el tercer ejemplo se sujeta un frasco por los extremos superior e inferior utilizando de nuevo tres dedos; y en el cuarto ejemplo se sujeta una botella de forma cilíndrica utilizando los cuatro dedos de la mano. Nótese que en los dos últimos ejemplos hay una notable diferencia entre las fuerzas producidas por los pares en las articulaciones y aquellas medidas en los sensores; esto se debe a que se está realizando fuerzas tangenciales sobre la superficie del sensor que este no detecta. Este efecto muestra la utilidad de la herramienta que rápidamente permite visualizar de forma gráfica esta diferencia, y por tanto lo que está pasando realmente con las fuerzas de presión.

5 CONCLUSIONES Y TRABAJO FUTURO

En este trabajo se ha presentado una herramienta para la visualización gráfica de las fuerzas que se generan debido al contacto entre las yemas de los dedos de una mano robótica y un objeto. Las fuerzas visualizadas provienen de dos fuentes: las fuerzas registradas por sensores táctiles ubicados en las yemas de los dedos de la mano robótica y las fuerzas calculadas a partir de los pares aplicados en las articulaciones de la mano. La herramienta también permite

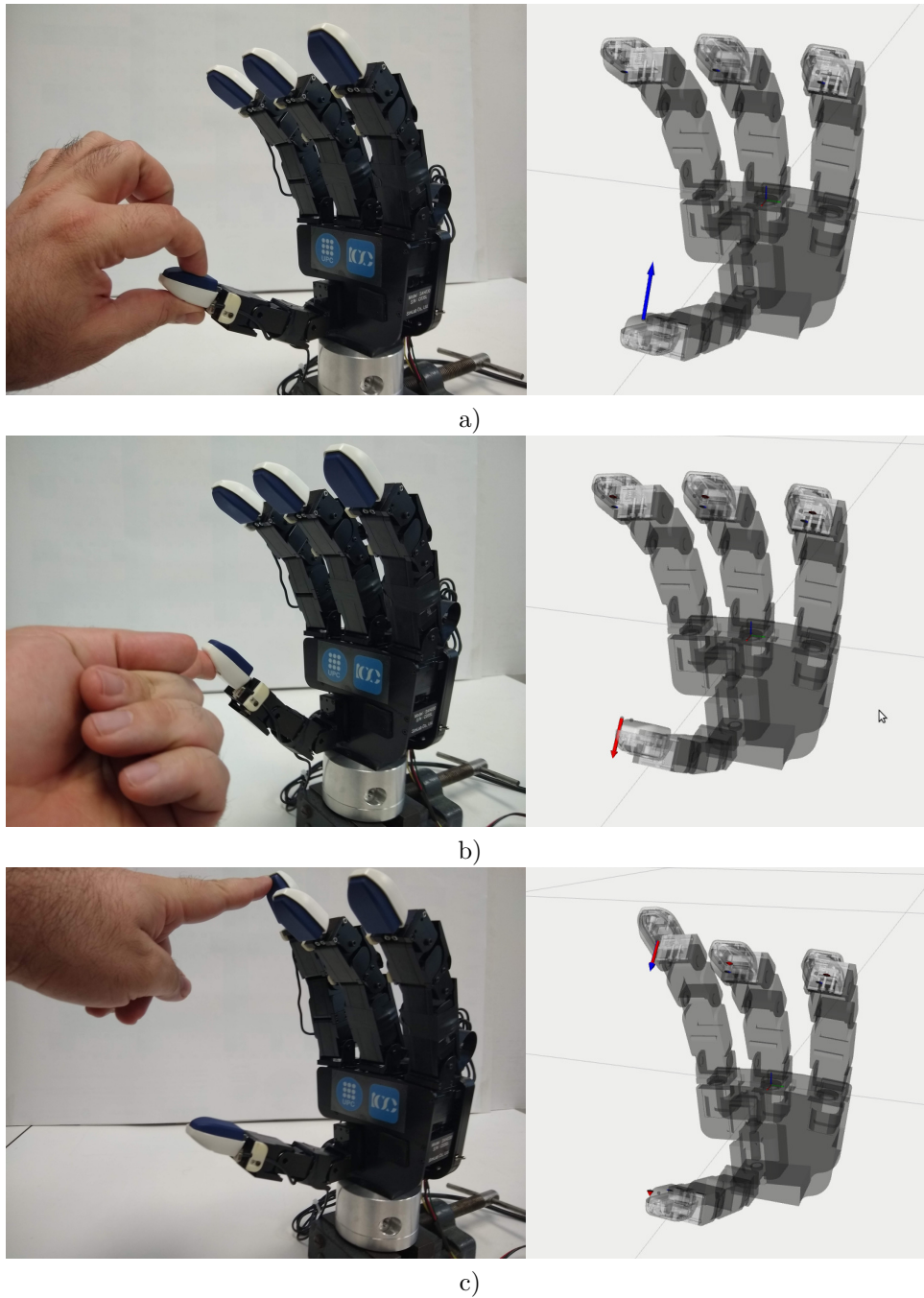


Figura 5: Ejemplos de interacción con los dedos y los sensores táctiles para obtener fuerzas que se visualizan en la interfaz gráfica de la herramienta. a) Fuerza registrada por el sensor táctil debido a una presión ejercida directamente sobre el sensor; b) Fuerza calculada a partir de los pares aplicados por el controlador para mantener la mano en una configuración estática cuando se aplica una fuerza sobre el dedo; c) Visualización simultánea de la fuerza registrada por el sensor y la fuerza calculada debido al par aplicado en las articulaciones cuando se ejerce una fuerza empujando el dedo desde la yema y en contacto con el sensor táctil.

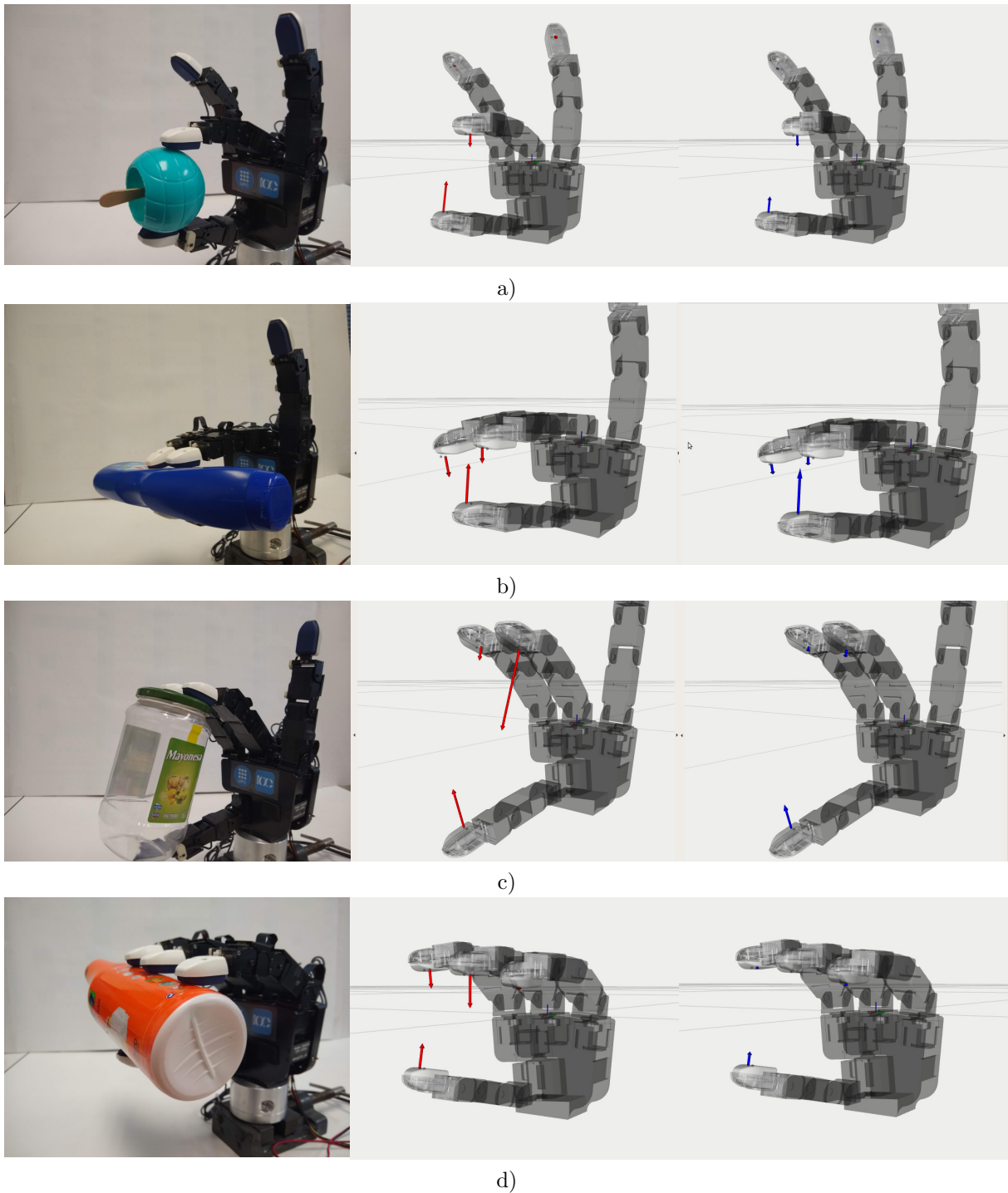


Figura 6: Ejemplos de prensiones de objetos con diferentes formas. a) Objeto con forma esférica sujetado con dos dedos; b) Botella con caras planas sujetado con tres dedos; c) Objeto sujetado por los extremos superior e inferior usando tres dedos; d) Botella de forma cilíndrica sujetada usando cuatro dedos.

controlar el movimiento individual de cada una de las articulaciones de la mano para ejecutar una prensión.

Como trabajo futuro se plantea la ampliación de la librería de archivos URDF introduciendo modelos de nuevas manos robóticas y sensores táctiles.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Gobierno Español a través del proyecto DPI2016-80077-R.

English summary

SOFTWARE TOOL FOR GRAPHICAL VISUALIZATION OF CONTACT FORCES AND MOVEMENTS OF A ROBOTIC HAND WITH TACTILE SENSORS

Abstract

In this paper we present a software tool that allows the graphical visualization of contact forces of special utility in grasping applications using robotic hands. The forces are generated both due to the contact between the fingertips and the manipulated object, or due to the torques applied in the joints. The tool also allows to move the real hand either by specifying configurations of the hand or by individual instructions for each joint, the movement is executed at the same time as it is visualized in the graphic simulation. As a practical example, an application using the Allegro hand with tactile sensors WTS-FT is presented and different grasping examples showing with the tactile information registered in the tactile sensors and the forces computed using the torques applied to the motors.

Keywords: Grasping, Manipulation, Software, Tactile sensors, Robotics.

Referencias

- [1] E. Coumans. Bullet - Real time physics simulation, 2015. URL <https://pybullet.org>.
- [2] N. T. Dantam, S. Chaudhuri, and L. E. Kavraki. The task-motion kit: An open

source, general-purpose task and motion-planning framework. *IEEE Robotics Automation Magazine*, 25(3):61–70, Sep. 2018.

- [3] D. Hershberger, D. Gossow, and J. Faust. rviz - 3D visualization tool for ROS, 2019. URL <http://wiki.ros.org/rviz>.
- [4] B. León, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moio, J. Bohg, J. Kuffner, and R. Dillmann. Opengrasp: A toolkit for robot grasping simulation. In *Proc. of the Second International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, SIMPAR, pages 109–120. Springer-Verlag, 2010.
- [5] A. T. Miller and P. K. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics Automation Magazine*, 11(4):110–122, 2004.
- [6] QT. Cross-platform software development for embedded & desktop, 2019. URL <https://www.qt.io/>.
- [7] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. Ros: an open-source robot operating system. In *Proc. of IEEE International Conference on Robotics and Automation – Workshop on Open Source Software*, pages 1–6, 2009.
- [8] J. Rosell, A. Pérez, A. Aliakbar, Muhayyuddin, L. Palomo, and N. García. The kautham project: A teaching and research tool for robot motion planning. In *Proc. of the IEEE Int. Conf. on Emerging Technologies and Factory Automation, ETFA'14*, 2014.
- [9] R. Smith. ODE: Open Dynamics Engine, Feb. 2006. URL <https://ode.org>.
- [10] I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012.
- [11] N. Vahrenkamp, M. Krohnert, S. Ulbrich, T. Asfour, G. Metta, R. Dillmann, and G. Sandini. Simox: A robotics toolbox for simulation, motion and grasp planning. *Advances in Intelligent Systems and Computing*, 1:585–594, 2013.



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>).