


## HACIA EL AGARRE DE OBJETOS UTILIZANDO APRENDIZAJE ROBÓTICO POR IMITACIÓN Y DATOS DE FUERZA

 CARLOS ANDRÉS PEÑA SOLÓRZANO <sup>1</sup>  
JOSÉ GABRIEL HOYOS GUTIÉRREZ <sup>2</sup>  
FLAVIO AUGUSTO PRIETO ORTIZ <sup>3</sup>

### RESUMEN

En este artículo se trata el agarre de objetos en robótica. Específicamente, agarres de precisión y la fuerza requerida en los puntos de contacto entre la mano y el objeto para realizar una buena sujeción. Se propone adquirir los datos de sensores de fuerza utilizando un guante de datos y codificándolos mediante aprendizaje por imitación. Se utilizan imágenes RGB y de profundidad para determinar la ubicación y orientación de los objetos. Se prueban varias configuraciones mano-objeto en simulación, comparando la calidad del agarre al utilizar las fuerzas máximas, mínimas y promedio truncado. La variación de la calidad obtenida es pequeña y en algunos casos despreciable, permitiendo concluir que al seleccionar siempre las fuerzas máximas, se obtiene un agarre que se ajusta bien a múltiples configuraciones. Además, se presenta un sistema de adquisición de datos de fuerza de bajo costo y una etapa de procesamiento de imágenes que permite determinar la ubicación y orientación de los objetos.

**PALABRAS CLAVE:** agarre; robótica; aprendizaje por imitación; programación por demostración; fuerza; procesamiento de imágenes.

## TOWARDS OBJECT GRASPING USING LEARNING BY IMITATION AND STRENGTH DATA

### ABSTRACT

This article deals with robotic object grasping. Specifically, precision grasps and the strength required in the contact points between the hand and the object to obtain a good grip. We propose to acquire the data of force sensors using a data glove and learning by imitation to encode it. RGB and depth images are used to determine objects location and orientation. Several hand-object configurations are simulated, comparing the grasp quality when maximum, minimum and truncated mean are used. The variation of grasp quality obtained is small and in some cases negligible, so we can conclude that by selecting the maximum grasping strength, we achieve a well-adjusted grasp to multiple configurations. Besides, we present a low cost strength acquisition system and an image processing stage which allows calculating the location and orientation of an object.

**KEYWORDS:** Grasping; Robotics; Learning by Imitation; Programming by Demonstration; Strength; Image Processing.

<sup>1</sup> Ingeniero electrónico Pontificia Universidad Javeriana. MCs en Ingeniería–Automatización Industrial, Universidad Nacional de Colombia. Asistente Editorial Revista Colombiana de Química, Universidad Nacional de Colombia.

<sup>2</sup> Ingeniero electrónico Universidad Tecnológica de Pereira. MCs. En Sistemas de Generación de Energía Eléctrica Universidad del Valle. PhD. Automática e Informática industrial Universidad Pontificia de Valencia, España. Profesor Titular, Universidad Nacional de Colombia.

<sup>3</sup> Ingeniero electricista Universidad Distrital Francisco José de Caldas. Licenciado en Física Universidad Nacional de Colombia sede Bogotá. MCs. en Ingeniería Eléctrica Universidad de los Andes. PhD. Institut National des Sciences Appliquées (INSA), Francia. PhD. École de Technologie Supérieure, Universidad de Québec. Docente, Universidad del Quindío.



*Autor de correspondencia:* Peña-Solórzano, C.A. (Carlos Andrés). Dirección postal: Cra. 45 No. 22-44 Apt. 104A, Bogotá, Colombia. Teléfono: (571) 6966151  
Correo electrónico: capenas@unal.edu.co

#### *Historia del artículo:*

Artículo recibido: 16-IX-2014 / Aprobado: 12-III-2015  
Disponible online: 30 de junio de 2015  
Discusión abierta hasta junio de 2016



## RUMO AO ADERÊNCIA DE OBJETOS UTILIZANDO APRENDIZAGEM ROBÓTICA POR IMITAÇÃO E DADOS DE FORÇA

### RESUMO

Em este artigo, se fala sobre os objetos de aderência em robótica é. Especificamente, apertos de precisão e a força necessária nos pontos de contato entre a mão e o objeto para um ajuste perfeito. Propõe-se a aquisição de dados de sensores de força, usando uma luva de dados e codificando-os por aprendizagem de imitação. São utilizadas imagens RGB e de profundidade para determinar a localização e a orientação de objetos. São testadas várias configurações de mão-objeto em simulação através da comparação da qualidade de aderência ao usar a força máxima, média, e mínimo. A variação na qualidade obtida é pequena e, em alguns casos negligenciável, levando à conclusão de que selecionando sempre as forças máximas, é obtido um aperto que se encaixa bem com várias configurações. Além disso, um sistema de aquisição de dados de força de baixo custo e uma etapa de processamento de imagem para determinar a posição e orientação de objetos.

**PALAVRAS-CHAVE:** Pegar; Robótica; A aprendizagem por imitação; A programação por demonstração; Força; Processamento de imagens.

### 1. INTRODUCCIÓN

Mientras que los robots industriales de hoy en día son capaces de resolver tareas específicas en ambientes conocidos, los robots que comparten su espacio de trabajo con los seres humanos tienen que trabajar en ambientes dinámicos que están diseñados para las necesidades humanas. Deben interactuar directamente con los seres humanos y tener la capacidad de realizar una gran variedad de tareas complejas. Además, ser lo suficientemente flexibles como para adaptarse fácilmente a nuevas tareas y a cambios inesperados en el entorno.

Reprogramar al robot para cada situación diferente es muy complejo, por lo que es importante para el área de investigación, desarrollar formas intuitivas de enseñarle (Grave, Stuckler, Behnke, 2010). Ello permitiría una mejora continua de las habilidades del robot para que pueda adaptarse fácilmente a nuevas situaciones. Una de las habilidades que debe ser perfeccionada en un robot es la de interactuar, por medio del agarre, con los objetos que están a su alrededor, debido a que se trata de una habilidad elemental para la interacción humano-robot. Las técnicas de entrenamiento para que el robot realice tareas complejas deben ser accesibles para cualquier persona, lo cual puede lograrse por medio del aprendizaje por imitación, también conocido como programación por demostración, donde una persona que no es experta en el tema puede enseñar habilidades complejas al robot, como las técnicas desarrolladas por Do, Asfour, y Dillmann (2011) y Kormushev, Calinon, y Caldwell (2011).

La adquisición de habilidades nuevas de agarre juega un papel esencial, para permitir que los robots humanoides puedan interactuar plenamente con el medio ambiente y el ser humano. Teniendo en cuenta la variedad y las diferentes maneras en que un objeto puede ser tratado, un simple cierre de la mano es insuficiente, debe realizarse una proyección completa de los posibles agarres asociados con cada objeto. Con el fin de iniciar este proceso, el aprendizaje por imitación es una alternativa rápida en términos de adquisición de nuevas habilidades.

Si un sistema robótico puede aprender a realizar una tarea de manipulación, mediante la observación de un maestro humano, el costo de la programación de sistemas flexibles y adaptables se puede reducir considerablemente (Bernardin, *et al.*, 2005). Para ello, el robot debe ser capaz de observar una demostración y deducir de ella una descripción simbólica de lo que se ha hecho. Esta representación de alto nivel puede ser utilizada para generar un plan de ejecución robusta, aun cuando las condiciones ambientales cambien.

Por otra parte, si una habilidad de agarre que es aprendida por el robot está basada en datos de movimiento, las restricciones de las tareas cinemáticas pueden ser extraídas de una serie de demostraciones que permiten su reproducción en objetos similares. Sin embargo, al aplicar la habilidad que fue aprendida por el robot en objetos de un tamaño o rigidez diferente, las fuerzas de interacción que resultarían podrían dañar la mano robótica o el obje-

to. Tales daños pueden evitarse considerando también la información de fuerza en el proceso de aprendizaje.

La metodología que se propone presenta los siguientes pasos: primero, la captura de la fuerza ejercida en la yema de los dedos debido al contacto con el objeto sujetado. Para esto, se construyó un guante de datos con sensores de fuerza en cada uno de los cinco dedos. Segundo, los perfiles de fuerza se codifican con la técnica de aprendizaje por imitación llamada primitivas de movimiento dinámico (DMP por sus siglas en inglés), disminuyendo la cantidad de memoria necesaria. Tercero, la reconstrucción (decodificación) de la señal sirve como entrada a la etapa de simulación de la sujeción en MatLab, donde se realizaron medidas de la calidad del agarre en diferentes configuraciones.

Adicionalmente, se muestran las etapas de procesamiento de imágenes que permiten determinar la orientación en el espacio de los objetos que se usaron durante la captura de fuerza.

El resto de este documento tiene la siguiente estructura: en la Sección 2 se presentan trabajos relacionados. En la Sección 3 se presenta la técnica propuesta, dividida en adquisición de los datos y teoría e implementación de la técnica de programación por demostración utilizada. Por último se presentan los resultados y las conclusiones.

## 2. TRABAJOS RELACIONADOS

Una de las técnicas de aprendizaje por imitación es las primitivas de movimiento dinámico (DMP), que permiten codificar los movimientos de trayectorias específicas del robot. En el caso de un movimiento discreto (punto a punto), cada grado de libertad del robot (DoF por sus siglas en inglés), se describe por un sistema de ecuaciones diferenciales no lineales (Kormushev, Calinon, Caldwell, 2011). Una de las propiedades de DMP es la habilidad integrada de reaccionar a perturbaciones, sin introducir discontinuidades en el movimiento resultante. Además, DMP no es directamente dependiente del tiempo, por lo que es fácil detener la ejecución del movimiento, sin tener en cuenta su evolución de forma extensiva.

Las DMPs pueden ser extendidas para incluir la capacidad de evitar obstáculos, cambiando de una primitiva a otra, mediante retroalimentación sensorial (Forte, Ude, Gams, 2011), o creando posiciones objetivo

virtuales, utilizando campos potenciales (Tan, *et al.*, 2011). Nemec y Ude (2012) proponen realizar acciones secuenciales, asegurando que las primitivas de movimiento consecutivas se unan en forma continua.

Además de replicar los movimientos o acciones aprendidas, utilizando una reconstrucción probabilística basada en regresión de procesos gaussianos (GPR por sus siglas en inglés), pueden crearse acciones para situaciones que el robot no ha encontrado anteriormente, lo cual se presenta tanto en simulación como en experimentos reales de alcanzar posiciones, tomar objetos y lanzar esferas (Ude, *et al.*, 2010; Forte, *et al.*, 2012).

Un factor importante para las técnicas de aprendizaje por imitación es el método de adquisición de los datos de la demostración. Como lo menciona Billard, *et al.* (2007), inicialmente se utilizaba teleoperación, siendo reemplazada progresivamente por la enseñanza cinestésica (guiando las partes del robot mediante movimiento de su propio cuerpo), e interfaces como guantes de datos, visión y sensores de distancia basados en láser.

Se ha difundido de manera amplia el aprendizaje robótico por imitación utilizando imágenes como datos de entrada (Azad, Asfour, Dillmann, 2007; León, *et al.*, 2011). Se pueden utilizar imágenes 2D, las cuales proveen información plana de la escena, e imágenes 3D, que contienen datos de profundidad. Desde el punto de vista de la ingeniería, las cámaras de video 2D presentan bajos costos, no son invasivas y proveen gran cantidad de información. Son especialmente útiles cuando se combinan con información del entorno o datos de profundidad (imágenes 3D) (Lopes y Santos-Victor, 2003).

En el pasado, los sensores que proveían imágenes 3D eran de difícil adquisición, dada su poca disponibilidad y altos precios. En Noviembre de 2010, Microsoft lanza el sensor Kinect, con la capacidad de producir flujos de imágenes RGB y de profundidad a un precio mucho menor a los sensores de rango tradicionales. Las imágenes de profundidad se obtienen de una cámara de tiempo de vuelo, que mide la distancia de cualquier punto desde el sensor. Una matriz de luz en el infrarrojo cercano se proyecta a través de la escena, y el sensor calcula el tiempo de vuelo que le toma, a cada punto de la matriz de luz generada, reflejarse desde el objeto (Kar, 2010).

Se pueden encontrar desarrollos en diversas áreas utilizando el sensor Kinect, entre éstos seguimiento en

3D de las articulaciones del esqueleto humano, o partes de éste (Kar, 2010; Oikonomidis, Kyriazis, Argyros, 2011). También se ha utilizado para reconocer gestos con la mano (Fрати y Prattichizzo, 2011), como sensor de contacto (Wilson, 2010) y como sensor de navegación y prevención de obstáculos (Fiala y Ufkes, 2011). Además, ha permitido la creación de interfaces de interacción humano-máquina más naturales (Villaroman, Rowe, Swan, 2011), a la vez que se presenta como una potencial tecnología para la educación interactiva (Hsu, 2011).

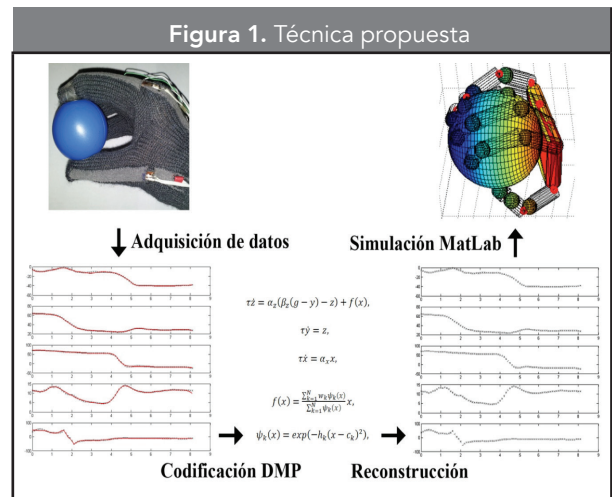
En cuanto a aprendizaje por imitación utilizando información de fuerza, se encuentran trabajos como Schmidts, Lee, y Peer (2011), donde se utiliza un dispositivo de captura de movimientos y fuerzas en la mano para agarrar un objeto simulado en un entorno virtual, a diferencia del trabajo que se propone en este documento, donde el agarre se realiza sobre un objeto real. En Rozo, Jiménez, y Torras (2010) y Rozo, Jiménez, y Torras (2011), se utiliza un dispositivo háptico para enseñar al robot la tarea, mientras el robot lee continuamente la información de fuerzas/torques relativas a la actividad. En el trabajo de Kormushev, Calinon, y Caldwell (2011), se utiliza tanto el dispositivo háptico para el aprendizaje de los perfiles de fuerza, como una enseñanza cinestésica para los perfiles de movimiento. En dichos trabajos, la fuerza no la realiza el instructor sobre el objeto directamente, como en el propuesto aquí, donde un guante de datos se utiliza para realizar el agarre, y la información obtenida se utiliza para la programación por demostración del robot.

### 3. TÉCNICA PROPUESTA

Como ya se mencionó, la técnica que se propone (Figura 1) inicia con la captura de la fuerza ejercida en la yema de los dedos debido al contacto con el objeto sujetado. Para esto, se construyó un guante de datos con sensores de fuerza en cada uno de los cinco dedos.

La información obtenida durante la adquisición se codifica con la técnica de aprendizaje por imitación llamada DMP, permitiendo pasar de 600 datos en memoria a sólo 20. La reconstrucción (decodificación) de la señal se utiliza como entrada a la etapa de simulación en MatLab, donde se mide la calidad del agarre en diferentes configuraciones, cuando se varía la fuerza ejercida por los dedos sobre el objeto.

Adicionalmente, se presentan las etapas para el cálculo de la orientación de los objetos usados du-



rante los experimentos, utilizando imágenes de color y profundidad.

#### 3.1. Adquisición de los datos

Esta sección se encuentra dividida en: datos de fuerza, adquiridos utilizando el guante de datos construido con los sensores ubicados en las yemas de los dedos, y datos de orientación de los objetos, donde se describen las etapas de procesamiento de imágenes que permiten determinar dos ángulos de orientación de tres objetos diferentes en el espacio.

##### 3.1.1. Datos de fuerza

El guante de datos consta de sensores de fuerza en la yema de los dedos, los cuales fueron preseleccionados según la disponibilidad. La selección final se realizó según criterios como el área de sensado, la sensibilidad, y el precio, como se observa en la **Tabla 1**. Se eligió el sensor FSR (*Force Sensitive Resistor*) de 0,5", cuyo rango de fuerza permite la lectura de datos entre 0 y 100 lb, y se propuso la manipulación de objetos con pesos en el rango estipulado.

Existen dos tipos de agarre: a) de fuerza, donde se utiliza la palma de la mano y el cierre de los dedos para sujetar el objeto y, b) de precisión, donde el agarre se realiza utilizando las yemas de los dedos. Se escogió este último como el tipo de sujeción a realizar, ya que es el tipo de presión más común en tareas de ensamble, en las que se pretende realizar desarrollos posteriores al trabajo presentado. Además, se definió el uso de cinco

sensores, ubicados en la punta de los dedos (**Figura 2**).

Las señales de los sensores de fuerza se acondicionan con un divisor de voltaje que lo protege de sobrecorrientes (**Figura 3**).

Para la adquisición de los datos se utiliza una tarjeta National Instruments NI-USB 6008.

### 3.1.2. Datos de orientación de los objetos

Los objetos seleccionados para el agarre son una esfera, un cubo y un cilindro. Para determinar la orientación de los objetos, se utilizó el sensor Kinect de Microsoft, el cual consta de una cámara RGB con resolución 640 x 480 y una cámara de profundidad basada en infrarrojo, obteniendo imágenes como las mostradas en las **Figuras 4a** y **4b**, respectivamente.

Para realizar la segmentación, se convirtió la imagen RGB a un mapa de colores HSV, donde se continuó trabajando con el canal de tono (hue), ya que es menos sensible a los cambios en iluminación. Con este canal, se encuentra el rango de valores que permiten segmentar el color azul de la esfera, obteniendo para el rango azul, una variación en el tono de:  $158 \leq \text{tono} \leq 166$ . La **Figura 5** muestra el resultado de la segmentación utilizando el rango definido.

La imagen segmentada presenta ruido, por lo cual se realiza una operación morfológica de apertura, la cual elimina secciones con áreas muy pequeñas. Luego, se utiliza el momento estadístico  $m_{00}$  del contorno resultante, el cual puede calcularse con la función momentos estadísticos, e indica directamente el área de dicho contorno. Finalmente, se realiza una operación donde sólo el contorno de mayor área permanece en la imagen final. La información obtenida se utiliza para segmentar la imagen de profundidad.

Posteriormente se calcula el centro del área que designa el objeto en la imagen, mediante el cálculo de los centroides en  $y$  y  $x$ , como los primeros momentos estadísticos sobre el área ( $\bar{x} = (m_{10}) / (m_{00})$  y  $\bar{y} = (m_{01}) / (m_{00})$ ). Para la imagen de ejemplo que se presenta, el valor corresponde a las coordenadas en píxeles: (280, 221).

La función *ProjectiveToRealWorld* de la librería de manejo del sensor Kinect, permite obtener a partir de los datos  $(x, y)$  en píxeles, los valores  $(X, Y, Z)$  del sistema 3D o sistema de coordenadas del mundo.

Tabla 1. Características utilizadas en la selección del sensor de fuerza

Característica	FSR Modelo #A201	FSR Modelo #A201	FSR 0,5" SEN-09375
Longitud total [mm]	197	25,4	60,3
Ancho total [mm]	14	14	19
Diámetro de detección[mm]	9,53	9,53	12,7
Rango de fuerza [lb]	0-100	0-100	0-25
Precio [pesos colombianos]	\$60 900	\$63 800	\$16 000

Figura 2. a) Ubicación de los sensores de fuerza. b) Agarre de la esfera con el guante de datos

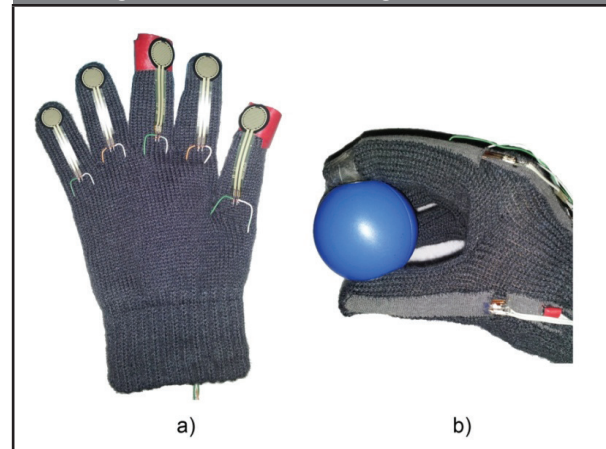
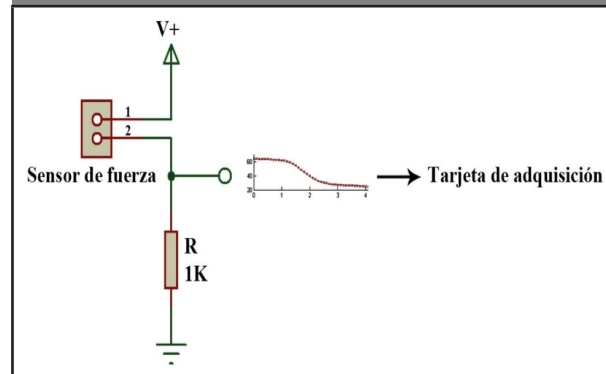
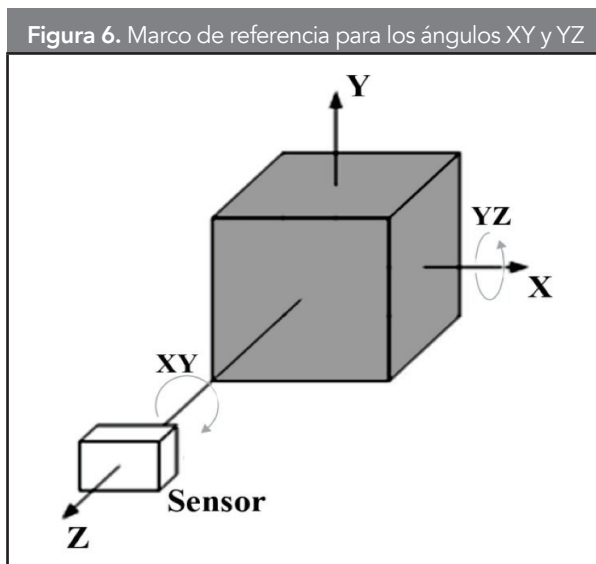
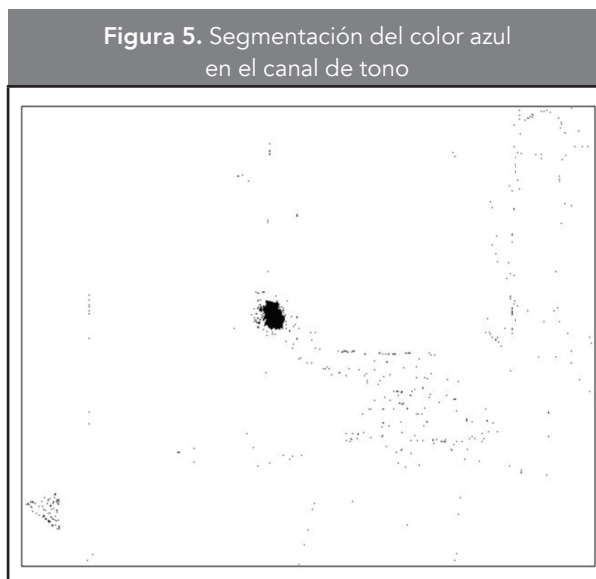
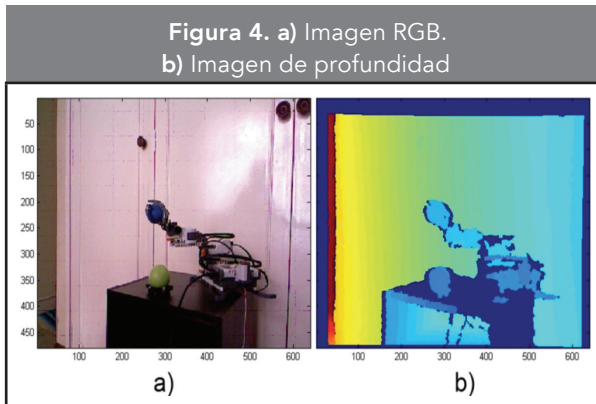


Figura 3. Circuito de acondicionamiento para cada uno de los sensores de fuerza





La matriz con las coordenadas del mundo consta de tres canales, cada uno con resolución 640×480. Para obtener los datos  $(X, Y, Z)$  en milímetros, se lee la matriz en la posición  $(u, v, Z)$ , donde  $u$  y  $v$  corresponden al centroide  $(x, y)$  obtenido anteriormente, y  $Z$  corresponde al valor de profundidad. Para la esfera azul en el ejemplo, dichos valores corresponden a las coordenadas en milímetros:  $(46, -26, -662)$ . Las operaciones anteriores permiten hallar la posición del objeto con respecto a la ubicación de la cámara, la cual corresponde al valor cero de los ejes de coordenadas  $(X, Y, Z)$ . Por tener simetría en ambos ejes, la esfera no requiere el cálculo de rotación.

Para facilitar el procesamiento de las imágenes con el cubo y el cilindro, se utilizaron marcadores en los objetos. En el caso del cubo, se marcó una línea en cada una de las caras de un color diferente al cuerpo del objeto. Mediante la segmentación de dicha línea (proceso análogo al procesamiento de la imagen con las esferas), y hallando el punto más alto y más bajo de esta, se tiene suficiente información para calcular la orientación del objeto en el espacio. La orientación se definió mediante dos ángulos, el ángulo  $XY$  y el ángulo  $YZ$  (**Figura 6**). El ángulo  $XY$  se refiere a la inclinación del objeto en un plano paralelo al plano de la cámara; es decir, el balanceo a izquierda o derecha con respecto a la imagen captada por la cámara (**Ecuación 1**). El ángulo  $YZ$  se refiere a la inclinación del objeto hacia la cámara; es decir, el balanceo adelante o atrás según la profundidad del punto superior con respecto al inferior (**Ecuación 2**).

$$\text{Tan}(XY) = \frac{X_1 - X_0}{Y_1 - Y_0} \quad (1)$$

$$\text{Tan}(YZ) = \frac{Y_1 - Y_0}{Z_1 - Z_0} \quad (2)$$

En el caso del cilindro, se marcaron las circunferencias superior e inferior con los colores azul y verde. Luego de segmentar ambos colores, se encuentra el punto más cercano a la cámara en cada contorno y, utilizando estos dos puntos, se calculan los ángulos de orientación de forma análoga al procedimiento de la imagen con el cubo. Para comparar los resultados, se midieron los ángulos reales, utilizando un *software* comercial de procesamiento de fotografías sobre varias imágenes obtenidas con cámaras diferentes a la utilizada en el proyecto, como se presenta en las **Tablas 2 y 3**. En ellas, el error se calculó como el valor real (*software*

**Tabla 2. Resultados del cálculo de la orientación del cubo**

Repetición	Cubo - Programa		Cubo - Real		error XY	error YZ
	XY	YZ	XY	YZ		
1	5,91	-2,42	3,8	-5,1	2,34 %	2,98 %
2	-6,19	14,3	-8,4	11,1	2,46 %	3,56 %
3	-12,51	-5,95	-13,7	-6,1	1,32%	0,17 %
4	10,00	-7,83	9,4	-13,9	0,67 %	6,74 %
5	-1,99	-1,07	-1,3	-1,3	0,77 %	0,26 %
6	-1,25	-9,07	-1,1	-10,8	0,17 %	1,92 %
7	9,83	-11,93	11,5	-16,3	1,86 %	4,86 %
8	6,49	-1,95	2,9	-4,8	3,99 %	3,17 %
9	0,79	-11,81	3,5	-17,8	3,01 %	6,66 %
10	-5,07	15,06	-5,6	11,8	0,59 %	3,62 %
			<b>Promedio</b>		<b>1,72 %</b>	<b>3,39 %</b>
			<b>Desviación</b>		<b>1,23 %</b>	<b>2,28 %</b>

**Tabla 3. Resultados del cálculo de la orientación del cilindro**

Repetición	Cilindro - Programa		Cilindro - Real		error XY	error YZ
	XY	YZ	XY	YZ		
1	-2,61	-5,73	-1,9	-4,6	0,79 %	1,26 %
2	4,06	12,68	2,0	11,3	2,29 %	1,53 %
3	15,75	5,84	12,8	4,3	3,28 %	1,71 %
4	2,49	10,78	3,0	13,3	0,57 %	2,80 %
5	-12,02	0,73	-12,7	3,7	0,76 %	3,30 %
6	4,68	8,66	6,9	5,7	2,47 %	3,29 %
7	9,47	18,66	8,3	18,0	1,30 %	0,73 %
8	9,10	0,71	9,0	1,3	0,11 %	0,66 %
9	3,67	4,36	4,0	1,9	0,37 %	2,73 %
10	1,27	-16,94	3,0	-19,3	1,92 %	2,62 %
			<b>Promedio</b>		<b>1,38 %</b>	<b>2,06 %</b>
			<b>Desviación</b>		<b>1,05 %</b>	<b>1,01 %</b>

de procesamiento de imágenes) menos el valor medido sobre el máximo valor posible, en este caso 90°. Con diez pruebas, el error para el cilindro se mantuvo inferior a 2,1 %, con una desviación estándar cercana al 1 %. En el caso del cilindro, el error es menor a 3,5 %, con desviación máxima de 2,3 %. Ambos errores se consideran bajos dada la actividad de agarre de objetos.

### 3.2. Primitivas de movimiento dinámico

Una primitiva de movimiento dinámico puede codificar una trayectoria específica de un robot. La trayectoria de cada grado de libertad dada en el espacio de articulaciones o de la tarea, está descrito por el siguiente sistema de ecuaciones diferenciales (Ude, *et al.*, 2010).

$$\tau \dot{z} = \alpha_z (\beta_z (g - y) - z) + f(x), \tag{3}$$

$$\tau \dot{y} = z, \tag{4}$$

$$\tau \dot{x} = \alpha_x x, \tag{5}$$

donde  $z$  es una variable auxiliar y  $x$  es la variable de fase utilizada para hacer implícita la dependencia de  $f$  del tiempo (con valor inicial  $x(0) = 1$ ).  $\alpha_x, \alpha_z, \beta_z$  y  $\tau$  son variables que deben definirse de tal forma que el sistema converja al punto de equilibrio  $(z, y, x) = (0, g, 0)$ . La función no lineal  $f$  contiene parámetros libres, que permiten al robot realizar un movimiento suavizado punto a punto de la posición inicial  $y_0$ , a la configuración final  $g$ .

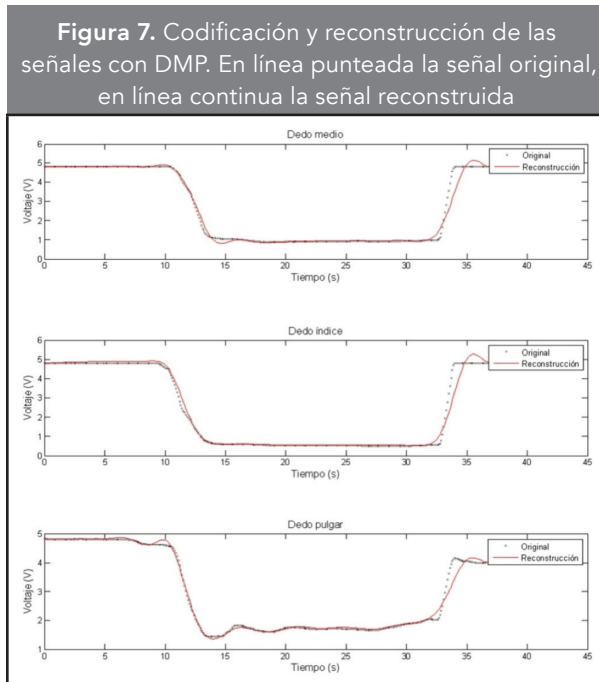
$$f(x) = \frac{\sum_{k=1}^N w_k \psi_k(x)}{\sum_{k=1}^N \psi_k(x)}, \tag{6}$$

$$\psi_k(x) = \exp(-h_k (x - c_k)^2), \tag{7}$$

donde  $\psi_k$  son las funciones de base radial,  $c_k$  son los centros de dichas funciones distribuidas en la trayectoria,  $w_k$  son los pesos y el ancho  $h_k > 0$ .  $w_k$  se estima para que el DMP codifique la trayectoria deseada. Cada grado de libertad está representado por su propio sistema de **Ecuaciones (3)-(4)**, con una fase común (5).

Antes de poder utilizar las ecuaciones de DMP para la codificación, es necesario ajustar los datos de entrenamiento, en este caso los datos de fuerza obtenidos con el guante de datos. El ajuste consiste en colocar las columnas de información según lo exige la librería utilizada.

Luego, es necesario definir los parámetros de la técnica ( $\alpha_x = 2, \alpha_z = 20, \beta_z = 5$ ), los cuales se proponen en la librería, pero pueden ajustarse para un mejor desempeño en la reconstrucción de las señales de entrada. Además de los parámetros, debe seleccionarse el número de funciones de base radial  $\psi_k$ , el cual debe ser de un tamaño adecuado para codificar correctamente las variaciones de la señal, manteniendo al mínimo



posible el número de variables en memoria. En este caso, se utilizó un número de funciones  $N = 10$ .

Finalmente, se convierten los datos de entrenamiento a DMP utilizando las **Ecuaciones (3)-(7)**, obteniendo los valores  $\tau$ ,  $g$  y  $w$ , que la definen y permiten la reconstrucción posterior de la señal.

La implementación de la técnica de aprendizaje por imitación se realizó en MatLab, utilizando una librería diseñada de forma específica. La librería de DMP, provee los algoritmos básicos que pueden utilizarse para estimar las primitivas de movimiento dinámico, a partir de una o múltiples demostraciones (Gams y Ude, 2009). Para ilustrar que la codificación disminuye la carga en memoria, no se requiere utilizar los datos arrojados por los sensores de fuerza de los cinco dedos, como sí se utilizan en el proceso de adquisición ilustrado en la sección 3.1.1. Sin embargo, a modo de ejemplo y para determinar el desempeño de la técnica de aprendizaje utilizada, se codificó la información obtenida de los sensores de los dedos pulgar, índice y medio. En la **Figura 7**, se presentan las señales de fuerza obtenidas, junto a las mismas señales, luego de ser codificadas y reconstruidas utilizando DMP.

Se codifica la información de fuerza utilizando DMP, obteniendo una reconstrucción posterior fiel a la

información de entrada, pasando de más de 600 datos por cada señal, a menos de 20 datos por cada primitiva de movimiento dinámico.

#### 4. RESULTADOS

A partir del guante de datos construido, se obtuvieron múltiples medidas al agarrar tres objetos mientras se giraba la mano. Los datos obtenidos varían según la fuerza ejercida sobre el objeto y la configuración mano-objeto. Una etapa de procesamiento de imágenes de varias fases fue implementada, la cual permite reconocer la orientación y posición en el espacio de los tres objetos seleccionados para el agarre, con respecto a la ubicación de la cámara.

Al no contar con una mano robótica, se simuló la sujeción con el toolbox de Matlab *SynGrasp* (Malvezzi, *et al.*, 2013), el cual permite simular objetos de diferentes formas y tamaños, rotados y ubicados en posiciones definidas por el usuario. Además, es posible determinar las fuerzas en los puntos de contacto entre mano y objeto, admitiendo análisis del mejor agarre gracias a indicadores de desempeño incluidos en el programa. El toolbox ha sido utilizado principalmente en desarrollos investigativos que involucran manos robóticas y sinergias posturales, donde a partir de unas cuantas variables de control pueden obtenerse la mayoría de los patrones de movimiento y configuración de la mano (Prattichizzo, Malvezzi, Bicchi, 2010; Gabiccini, *et al.*, 2011; Guo, *et al.*, 2013).

La función de medida de calidad del agarre se denomina *uniformidad de transformación* ( $Ut$ ), y fue seleccionado porque relaciona datos obtenidos de los dedos con el efecto que produce sobre el objeto que se sujeta (**Ecuación 8**).

$$Ut = \eta_c(H) = \frac{\sigma_{max}(H)}{\sigma_{min}(H)}, \quad (8)$$

Donde  $\eta_c(H)$  es el número de condición del Jacobiano de un dedo, mientras que  $\sigma_{max}$  y  $\sigma_{min}$  son sus valores singulares máximo y mínimo, respectivamente. La transformación entre el dominio de velocidades de las articulaciones de los dedos y este mismo dominio para el objeto, se considera uniforme cuando la contribución de las velocidades de cada articulación es igual en todas las componentes de la velocidad del objeto. Este concepto puede utilizarse para medir la calidad



del agarre, como lo determinan Roa, Suárez, y Cornelia (2008), y Veber, Dolanc, y Bajd (2005).

Los objetos utilizados en la simulación son similares a los utilizados en la realidad durante el agarre con el guante de datos (esfera, cilindro y cubo) (**Figura 8**). Se realizaron 40 tomas de datos de fuerza por objeto (obteniendo una desviación menor a 5,5 %), se determinaron los valores máximos, mínimos y promedios truncados (donde se promedia sin tener en cuenta los valores máximo y mínimo), de cada dedo. Esta información se codifica con DMP y la reconstrucción posterior se lleva a la librería de simulación en MatLab, donde se compara la calidad de los distintos agarres al variar los valores de fuerza.

Además, para cada objeto se simularon cuatro agarres diferentes, donde cambia la ubicación de la mano sobre el objeto, definidos como inferior, superior, frontal y lateral. El inferior sostiene el objeto cuando la mano se encuentra debajo de este, mientras el superior presenta la mano posicionada sobre el objeto. La sujeción frontal ubica la mano frente al objeto, con el dedo pulgar bajo este y los demás dedos encima. El lateral presenta la mano frente al objeto, pero con una rotación de 90°, ubicando los dedos a los lados del objeto.

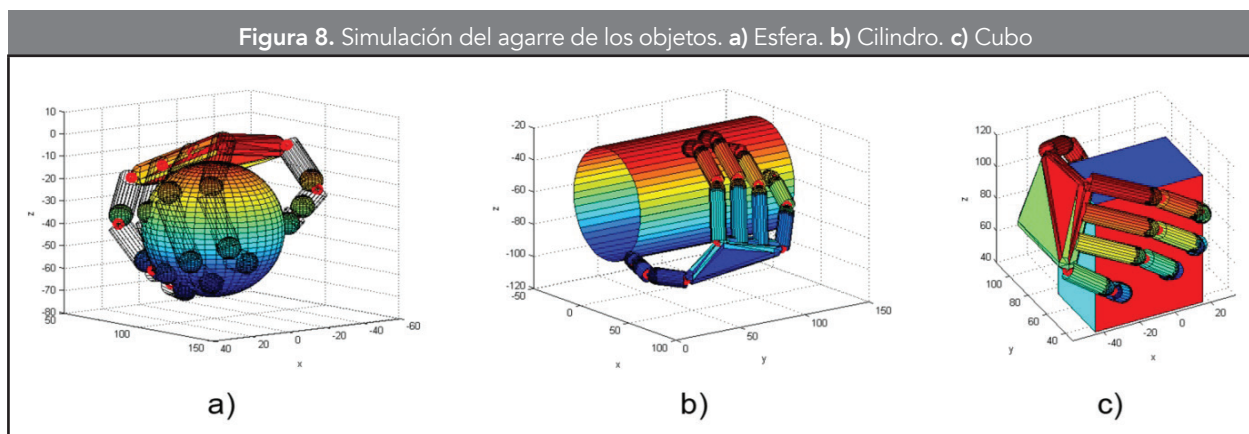
Teniendo en cuenta que la medida de uniformidad de transformación representa mayor calidad del agarre cuando el resultado es cercano a 1, se toma el valor obtenido que cumple dicha característica y se utiliza como base de comparación para cada tipo de sujeción de cada objeto.

La **Tabla 4** presenta los resultados obtenidos al sujetar los tres objetos, cada uno con cuatro

configuraciones de agarre. Aparece como cero, el valor de uniformidad de transformación más cercano a uno, y en las demás celdas, los porcentajes de variación con respecto a dicho valor. Cada configuración es comparada independientemente; en cada fila se presenta en negrilla el valor que se utilizó para comparar los valores obtenidos en dicha fila. En la **Tabla 4**, se utilizan porcentajes de variación para mantener las comparaciones sólo entre los valores indicados en cada fila, y evitar símiles entre distintas configuraciones u objetos que no deben realizarse en este contexto.

En el caso de la esfera, aunque los datos tomados con el guante corresponden a un agarre de precisión, es decir, tocando el objeto únicamente con la punta de los dedos, la simulación solo pudo realizarse con un agarre de fuerza, donde los dedos aprisionan el objeto contra la palma de la mano. La librería utilizada realiza este agarre por defecto cuando el objeto simulado es una esfera, y no pudo modificarse este comportamiento. Se contaba con los datos de la sujeción de precisión para este objeto, así que se realizó la simulación, aun cuando no hay correspondencia entre los tipos de agarre.

Durante la toma de datos con el guante, el movimiento de la muñeca asegura que la fuerza sobre el objeto varíe debido a la aceleración de la gravedad actuando sobre el objeto y modificando la fuerza de fricción. Este comportamiento se observa en la simulación del agarre con el cubo y el cilindro, donde los valores de uniformidad de transformación cambian para cada configuración de agarre y para los cambios en la fuerza ejercida. En el caso de la simulación con la esfera, el agarre de fuerza contrarresta la aceleración



**Tabla 4.** Medida de la calidad del agarre: uniformidad de transformación

Objeto	Configuración	Máxima fuerza (variación)	Mínima fuerza (variación)	Fuerza promedio truncada (variación)
Esfera	Inferior	3,46 %	1,77 %	0
	Superior	3,46 %	1,77 %	0
	Frontal	3,46 %	1,77 %	0
	Lateral	3,46 %	1,77 %	0
Cilindro	Inferior	0,02 %	0,01 %	0
	Superior	0,12 %	0	0,02 %
	Frontal	0	0,02 %	0,11 %
	Lateral	0	0,04 %	0,00 %
Cubo	Inferior	0	0,06 %	0,02 %
	Superior	0,04 %	0	0,04 %
	Frontal	0	0,03 %	0,02 %
	Lateral	0,01 %	0	0,03 %

de la gravedad, obteniendo los mismos valores de  $U_t$  para cada configuración de sujeción.

Para la esfera, se observa que la diferencia entre las  $U_t$  obtenidas tiene una variación mayor al compararla con los demás objetos, debido a la fuerza ejercida por los puntos de contacto de la mano que no se tienen en cuenta. Con esta información puede concluirse que para las actividades de sujeción, es importante tener en cuenta si un agarre preciso o fuerte se ajusta mejor a la situación u objeto. Además, la información de fuerza utilizada durante la simulación debe provenir de fuerzas de entrenamiento obtenidas al realizar la sujeción adecuada.

## 5. CONCLUSIONES

Se utilizó la técnica de aprendizaje por imitación, primitivas de movimiento dinámico, para la codificación y posterior reconstrucción de las fuerzas obtenidas al sujetar diferentes objetos. Se simularon diferentes configuraciones mientras se medía la calidad del agarre. La selección de la fuerza a utilizar depende tanto de la configuración de los dedos, como de la posición de estos sobre el objeto (la importancia viene dada por la acción

de la gravedad). Aun así, la variación de la calidad del agarre entre las fuerzas máximas, mínimas y promedio truncado es pequeña y en algunos casos despreciables, permitiendo concluir que al seleccionar siempre las fuerzas máximas, se obtiene una sujeción que se ajusta bien a múltiples configuraciones.

Utilizando imágenes RGB junto a imágenes de profundidad, puede determinarse la orientación de los objetos, realizando una etapa de segmentación y selección de puntos de interés, seguido de una etapa de cálculo que determina los ángulos entre los puntos encontrados y un marco de referencia definido.

Se plantea como trabajo futuro realizar pruebas con agarre de fuerza, teniendo en cuenta la fuerza en el punto de contacto de la palma de la mano. Se propone también la toma de datos mientras se realizan cambios en la configuración de la sujeción y se utilizan objetos con diferentes texturas, lo que conlleva a cambios en la fricción entre la mano y el objeto. Además, utilizar la información de la orientación de los objetos, para estimar y reproducir el agarre con un brazo manipulador robótico.

## AGRADECIMIENTOS

Los autores agradecen a Colciencias - Programa Jóvenes Investigadores e Innovadores, convocatoria 617 de 2013, por la financiación de este proyecto.

## REFERENCIAS

- Asad, P., Asfour, T.; Dillmann, R. (2007). Toward an Unified Representation for Imitation of Human Motion on Humanoids. In *2007 IEEE International Conference on Robotics and Automation*, pp. 2558 –2563.
- Bernardin, K.; Ogawara, K.; Ikeuchi, K.; Dillmann, R. (2005). A Sensor Fusion Approach for Recognizing Continuous Human Grasping Sequences Using Hidden Markov Models. *IEEE Transactions on Robotics*, 21(1), pp. 47–57.
- Billard, A.; Calinon, S.; Dillmann, R.; Schaal, S. (2007). Handbook of Robotics Chapter 59: Robot Programming by Demonstration. *Robotics*, chapter 59, pp. 1371–1394.
- Do, M.; Asfour, T.; Dillmann, R. (2011). Towards a Unifying Grasp Representation for Imitation Learning on Humanoid Robots. In *2011 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 482–488.
- Fiala, M.; Ufkes, A. (2011). Visual Odometry Using

- 3-Dimensional Video Input. In *2011 Canadian Conference on Computer and Robot Vision (CRV)*, pp. 86–93.
- Forte, D.; Gams, A.; Morimoto, J.; Ude, A. (2012). On-line Motion Synthesis and Adaptation Using a Rrajjectory Database. *Robotics and Autonomous Systems*, 60(10), pp. 1327–1339.
- Forte, D.; Ude, A.; Gams, A. (2011). Realtime Generalization and Integration of Different Movement Primitives. In *2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 590–595.
- Fрати, V.; Prattichizzo, D. (2011). Using Kinect for Hand Tracking and Rendering in Wearable Haptics. In *2011 IEEE World Haptics Conference (WHC)*, pp. 317–321.
- Gabiccini, M.; Bicchi, A.; Prattichizzo, D.; Malvezzi, M. (2011). On the Role of Hand Synergies in the Optimal Choice of Grasping Forces. *Autonomous Robots*, 31(2-3), pp. 235–252.
- Gams, A.; Ude, A. (2009). Generalization of Example Movements with Dynamic Systems. In *2009 9th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 28–33.
- Grave, K.; Stuckler, J.; Behnke, S. (2010). Improving Imitated Grasping Motions Through Interactive Expected Deviation Learning. In *2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, pp. 397–404.
- Guo, D.; Sun, F.; Zhang, J.; Liu, H. (2013). A Grasp Synthesis and Grasp Synergy Analysis for Anthropomorphic Hand. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1617–1622.
- Hsu, H. J. (2011). The Potential of Kinect as Interactive Educational Technology. In *2011 2<sup>nd</sup> International Conference on Education and Management Technology IPCSIT*, 13, pp. 334–338.
- Kar, A. (2010). Skeletal Tracking Using Microsoft kinect. *Methodology*, 1, pp. 1–11.
- Kormushev, P.; Calinon, S.; Caldwell, D. G. (2011). Imitation Learning of Positional and Force Skills Demonstrated Via Kinesthetic Teaching and Haptic Input. *Advanced Robotics*, 25(5); pp. 581–603.
- León, A.; Morales, E.; Altamirani, L.; Ruiz, J. (2011). Teaching a Robot New Tasks Through Imitation and Feedback. In *Proc. Workshop: New Developments in Imitation Learning, as part of the International Conference on Machine Learning*.
- Lopes, M.; Santos-Victor, J. (2003). Visual Transformations in Gesture Imitation: What you See is What you Do. In *IEEE International Conference on Robotics and Automation, 2003. ICRA '03.*, 2, pp. 2375 – 2381.
- Malvezzi, M.; Gioioso, G.; Salvietti, G.; Prattichizzo, D.; Bicchi, A. (2013). Syngrasp: A Matlab Toolbox for Grasp Analysis of Human and Robotic Hands. In *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1088–1093.
- Nemec, B.; Ude, A. (2012). Action Sequencing Using Dynamic Movement Primitives. *Robotica*, 30(5), 837.
- Oikonomidis, I.; Kyriazis, N.; Argyros, A. (2011). Efficient Model-Based 3d Tracking of Hand Articulations Using Kinect. In *BMVC 2011*.
- Prattichizzo, A. B. D.; Malvezzi, M.; Bicchi, A. (2010). On Motion and Force Control of Grasping Hands with Postural Synergies. *2010 Robotics: Science and Systems Conference*, pp. 27–30.
- Roa, M.; Suárez, R.; Cornelia, J. (2008). Medidas de calidad para la prensión de objetos. *Revista Iberoamericana de Automática e Informática Industrial (RIAI)*, 5(1), 66–82.
- Rozo, L.; Jiménez, P.; Torras, C. (2011). Robot Learning from Demonstration of Force-Based Tasks with Multiple Solution Trajectories. In *2011 15th International Conference on Advanced Robotics (ICAR)*, pp. 124–129.
- Rozo, L. D.; Jiménez, P.; Torras, C. (2010). Learning Force-Based Robot Skills from Haptic Demonstration. In *CCIA*, pp. 331–340.
- Schmidts, A. M.; Lee, D.; Peer, A. (2011). Imitation Learning of Human Grasping Skills from Motion and Force Data. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1002–1007.
- Tan, H.; Erdemir, E.; Kawamura, K.; Du, Q. (2011). A Potential Field Method-Based Extension of the Dynamic Movement Primitive Algorithm for Imitation Learning

**PARA CITAR ESTE ARTÍCULO /  
TO REFERENCE THIS ARTICLE /  
PARA CITAR ESTE ARTIGO /**

Peña-Solórzano, C.A.; Hoyos-Gutiérrez, J.G.; Prieto-Ortiz, F.A. (2015). Hacia el agarre de objetos utilizando aprendizaje robótico por imitación y datos de fuerza. *Revista EIA*, 12(23) enero-junio, pp. 71-82. [Online]. Disponible en: <http://dx.doi.org/10.14508/reia.2015.12.23.71-82>.

- with Obstacle Avoidance. In *2011 International Conference on Mechatronics and Automation (ICMA)*, pp. 525–530.
- Ude, A.; Gams, A.; Asfour, T.; Morimoto, J. (2010). Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives. *IEEE Transactions on Robotics*, 26(5), pp. 800–815.
- Veber, M.; Dolanc, M.; Bajd, T. (2005). Optimal Grasping in Humans. *Journal of Automatic Control*, 15, suppl., pp. 15-18.
- Villaroman, N.; Rowe, D.; Swan, B. (2011). Teaching Natural User Interaction Using OpenNI and the Microsoft Kinect Sensor. In *Proceedings of the 2011 conference on Information technology education*, pp. 227–232.
- Wilson, A. D. (2010). Using a Depth Camera as a Touch Sensor. In *ACM International Conference on Interactive Tabletops and Surfaces, ITS '10*, pp. 69–72, New York, NY, USA.