

Mínimos Cuadrados Recursivos para un Manipulador que Aprende por Demostración

José de Jesús Rubio^{a,*}, Enrique García^a, Gustavo Aquino^a, Carlos Aguilar-Ibáñez^b, Jaime Pacheco^a, Jesús A. Meda-Campaña^c

^aSección de Estudios de Posgrado e Investigación, Esime Azcapotzalco, Instituto Politécnico Nacional, Av. de las Granjas no. 682, México D.F., 02250, México.

^bCentro de Investigación en Computación, Instituto Politécnico Nacional, Av. Juan de Dios Bátiz S/N, México D.F., 07738, México.

^cSección de Estudios de Posgrado e Investigación, Esime Zacatenco, Instituto Politécnico Nacional, Av. IPN S/N, México D.F., 07738, México.

Resumen

En este trabajo, se desarrolla un sistema de control automatizado para permitir que un manipulador aprenda y planifique las trayectorias a partir de las demostraciones dadas por la mano de un usuario. La entrada de datos es adquirida por un sensor, y se aprende su comportamiento a través de un algoritmo de aprendizaje automático basado en los mínimos cuadrados recursivos. Se utiliza un perfil de trayectoria de interpoladores a tramos para evitar el movimiento impulsivo del manipulador. Se realiza el análisis de las cinemáticas directa e inversa para obtener los valores de las variables articulares para el manipulador. Se crea un modelo dinámico usando la formulación de Newton-Euler. Se aplica un control proporcional derivativo al sistema. Los sistemas de monitoreo y control se implementan en una plataforma embebida para fines de prueba.

Palabras Clave:

Manipulador, mínimos cuadrados recursivos, trayectoria, cinemática, modelo, plataforma embebida

Recursive Least Squares for a Manipulator which Learns by Demonstration

Abstract

In this work, a control system is developed to allow a manipulator to learn and plan trajectories from demonstrations given by the hand of an user. The input of data is acquired by a sensor, and its behavior is learned through an automatic learning algorithm based on the recursive least squares. A trajectory profile of interpolators to stretches is used to avoid the impulsive jerk on manipulators motion. Direct and inverse kinematics analysis is done for obtaining the joints variables values of the manipulator. A dynamic model is created using Newton-Euler formulation. A proportional derivative control is applied to the system. The monitoring and control systems are implemented in an embedded platform for testing purposes.

Keywords:

Manipulator, recursive least squares, trajectory, kinematics, model, embedded platform

1. Introducción

Actualmente existen muchos tipos de manipuladores que realizan tareas industriales y que no necesariamente tienen que ser programados de manera explícita para que realicen dichas actividades. El incremento en las habilidades de re-programación de forma no explícita en los robots manipuladores les otorga mayor autonomía por que les permite interactuar de mejor manera con el ambiente que les rodea. Debido a esto, los manipuladores programados por tele-operación y por comandos gestuales de movimiento están incrementándose, debi-

do a la necesidad de tener manipuladores genéricos que puedan llevar a cabo múltiples tareas y que se puedan adaptar a un ambiente humano. La ventaja de usar la mano de un usuario es la habilidad que tiene para ejecutar tareas heterogéneas en una amplia variedad de objetos. Esto sugiere que el cerebro humano desarrolla de una manera eficiente la capacidad de interactuar con objetos en espacios multidimensionales y que al ser puestos como referencia para ser copiados por sistemas robóticos, esto se complica ya que se deben utilizar algoritmos de planeación de movimiento también heterogéneos.

* Autor para correspondencia: jrubioa@ipn.mx; rubio.josedejesus@gmail.com

Existen algunos estudios interesantes realizados a controles aplicados a sistemas robóticos. En Calderón et al. (2017), Li et al. (2017), Rosado et al. (2017), y Serrano et al. (2018) se diseñan los controles para las prótesis robóticas usadas en la rehabilitación. En García et al. (2017), Liu et al. (2017), Pan et al. (2016), Sa et al. (2017), y Sun et al. (2017) se diseñan controles aplicados robots manipuladores. En Hernández et al. (2017), Marín et al. (2014), Marín et al. (2013), Ortigoza et al. (2016), Serrano et al. (2018), y Zhang et al. (2014) se estudia el seguimiento de la trayectoria en los robots móviles. En Hernández et al. (2016), Olivares et al. (2014), Páramo et al. (2017), Peng et al. (2017), y Rubio (2018) se enfocan en los controles aplicados a péndulos. Los trabajos mencionados demuestran que los controles aplicados a los sistemas robóticos son un tema de actual interés y estudio.

Una estrategia para llevar a cabo una tarea es la de imitar el comportamiento de la mano de un usuario a partir de la observación. La solución propuesta en este trabajo es la de utilizar un sensor de profundidad junto con un algoritmo de aprendizaje automático para que, a partir de la enseñanza de un usuario, el manipulador pueda llevar a cabo acciones repetitivas que el usuario le enseña con el movimiento de su mano. El aprendizaje por imitación es una capacidad que tienen los seres humanos para adquirir habilidades nuevas; también es una nueva área que permitirá a los robots aprender a través de la enseñanza. Una ventaja a nivel industrial, es que el aprendizaje por imitación no requiere que el ser humano maestro tenga conocimientos de programación.

En general los modelos de aprendizaje automático con sistemas robóticos son conocidos como la mejor trayectoria que predice un comportamiento real a partir de los datos experimentales. En Candelas et al. (2014), Espinosa et al. (2018), Martín et al. (2018), Méndez et al. (2011), Pomares et al. (2012), y Torres et al. (2009) se considera la generación de trayectorias y guiado mediante visión. En el caso en el que existen múltiples factores que afectan la salida del sistema y teniendo como objetivo la aproximación con exactitud, se utiliza la regresión lineal multivariable. En Ngoc et al. (2018), Rubio et al. (2013), Rubio (2017), y Vázquez et al. (2012) se propone el método de las pendientes para modelar señales cerebrales obtenidas experimentalmente, este método ha mostrado mejorar el desempeño en el entrenamiento de comportamientos no lineales que otros métodos matemáticos como el de k vecinos más cercanos Kanj et al. (2016). El método de las pendientes ha demostrado tener buenos resultados en el modelado de sistemas con comportamientos no lineales. Sin embargo presenta problemas, principalmente con los bordes de las funciones que componen el modelo, ya que al ser una técnica que utiliza rectas para aproximar una señal, es inevitable la formación de vértices en los extremos de las funciones, generando así una curva que no es derivable en estos puntos (suave a pedazos). Por lo cual, surge la necesidad de proponer un algoritmo que pueda aprender el comportamiento de los sistemas no lineales y genere una trayectoria suave.

La primer contribución principal de este documento es la de proponer un modelo de aprendizaje automático de los sistemas no lineales. El diseño del modelo de aprendizaje automático propuesto tiene las siguientes dos etapas. Primero, se describe el modelo de aprendizaje automático para el aprendizaje de

sistemas no lineales, donde éste considera el caso de múltiples entradas y múltiples salidas. Y segundo, se diseña el algoritmo de los mínimos cuadrados recursivos para conseguir el aprendizaje automático recursivo de los sistemas no lineales.

La segunda contribución principal de este documento es la de proponer un perfil de trayectoria de interpoladores a tramos en la cual, la modificación consiste en incluir un parámetro constante conocido como número de periodos transitorios que puede ser seleccionado por el usuario tal que se suavice la trayectoria deseada, y a su vez, el evitar algún movimiento brusco del manipulador.

La tercera contribución principal de este documento es la de sugerir la dinámica del sistema propuesto, la cual consiste en la obtención de las cinemáticas directa e inversa, del modelo dinámico, y del control de un manipulador de seis grados de libertad.

La última contribución principal de este documento es el desarrolló un sistema de control automatizado experimental que utiliza el aprendizaje automático, el perfil de trayectoria, y la dinámica para permitir que un manipulador aprenda y planifique las trayectorias a partir de las demostraciones dadas por la mano de un usuario.

El artículo está estructurado de la siguiente manera. En la sección 2 se aborda el desarrollo del control para el robot manipulador que aprende por demostración. En la sección 3 se realiza una comparación entre los controladores aplicados al robot manipulador en la simulación y en la experimentación. Finalmente, en la sección 4 se explican las conclusiones pertinentes y se mencionan los trabajos futuros relacionados con el tema.

2. El control para el manipulador que aprende por demostración

La Figura 1 muestra todos los elementos que permiten el control del manipulador. El objetivo de esta propuesta es que el robot manipulador va a realizar el movimiento llevado a cabo por la mano de un usuario. El proceso se describe a continuación. La mano de un usuario especifica la trayectoria deseada, para lo cual, el sensor toma los datos del movimiento de la mano de un usuario. Los movimientos percibidos por el sensor, son aprendidos por el algoritmo de aprendizaje automático mediante los mínimos cuadrados recursivos para generar una trayectoria deseada en el espacio tridimensional. Después, esta trayectoria es suavizada con el perfil de trayectoria de interpoladores a tramos. Posteriormente, esta trayectoria tridimensional suave se usa en el modelo cinemático inverso junto con el modelo dinámico para que a partir de la trayectoria tridimensional se generen las trayectorias deseadas en cada una de las articulaciones del robot manipulador. El controlador proporcional derivativo (PD) en conjunto con la instrumentación se usan para que cada una de las articulaciones del robot manipulador sigan a las trayectorias deseadas.

Las siguientes sub-secciones describen cada uno de estos elementos, los cuales se dividen en dos partes importantes: el aprendizaje automático y la dinámica del sistema.

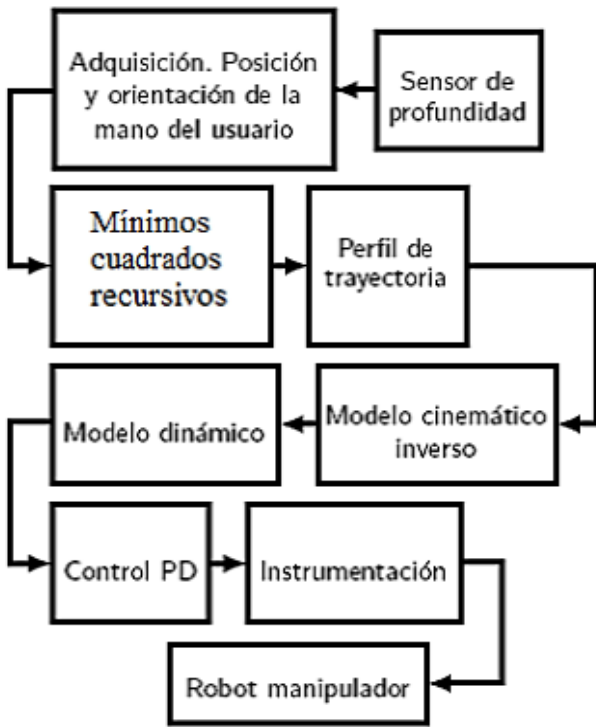


Figura 1: Diagrama de flujo del sistema de control

2.1. El aprendizaje automático

El algoritmo de aprendizaje automático y supervisado se usa para predecir una salida y a partir de ejemplos previos \mathbf{X} . Los modelos de regresión son conocidos como la mejor curva que predice un valor real desde los datos. En el caso de que hay factores múltiples que afectan la salida y con el objetivo de aproximar con exactitud se usa una regresión no lineal multivariable. \mathbf{X} representa los datos de entrada, y representa la salida, entonces:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{21} & \dots & x_{d1} \\ x_{12} & x_{22} & \dots & x_{d2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1(m-1)} & x_{2(m-1)} & \dots & x_{d(m-1)} \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{(m-1)} \end{bmatrix} \quad (1)$$

donde d representa el número de entradas o características y $m - 1$ representa el número de ejemplos de entrenamiento. Para la regresión no lineal el modelo de predicción es $\widehat{\mathbf{y}} = \mathbf{ax} + b$, pero si \mathbf{w} representa los pesos $\mathbf{w} = [w_1 \ w_2 \ \dots \ w_d]$ y \mathbf{x}_i representa un vector de características $\mathbf{x}_i = [x_{1i} \ x_{2i} \ \dots \ x_{di}]^T$ ambos con tamaño d , entonces el modelo de regresión simple se convierte en $\widehat{y}_i = \mathbf{wx}_i + b$. Si $w_0 = b$ y $x_0 = 1$ se puede reescribir \mathbf{w} como $\mathbf{w} = [w_0 \ w_1 \ w_2 \ \dots \ w_d]$ y \mathbf{x}_i como $\mathbf{x}_i = [x_0 \ x_{1i} \ x_{2i} \ \dots \ x_{di}]^T$, entonces el modelo de predicción con la regresión no lineal múltiple se convierte en:

$$\widehat{\mathbf{y}}_i = \mathbf{wx}_i \quad (2)$$

donde $\mathbf{w} = [w_0 \ w_1 \ w_2 \ \dots \ w_d]$ son los pesos y $\mathbf{x}_i = [x_0 \ x_{1i} \ x_{2i} \ \dots \ x_{di}]^T$ son las características.

Se define el error de aprendizaje e_i como sigue:

$$e_i = \widehat{y}_i - y_i = \mathbf{wx}_i - y_i \quad (3)$$

donde y_i es la salida (1), \widehat{y}_i es la predicción con la regresión no lineal múltiple (2).

La función de costo J se define como sigue:

$$J = \sum_{i=0}^{m-1} e_i^2 \quad (4)$$

donde e_i es el error de aprendizaje (3).

Para obtener la mejor predicción \widehat{y}_i para la salida y_i , se requiere encontrar el mínimo error e_i^2 , este objetivo se logra al encontrar los pesos \mathbf{w} que permiten lograr el mínimo de la función de costo J , que se representa como sigue:

$$\frac{\partial J}{\partial \mathbf{w}} = 0 \quad (5)$$

Realizando las operaciones para obtener (5), se obtiene:

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{w}} &= \sum_{i=0}^{m-1} \frac{\partial J}{\partial e_i} \frac{\partial e_i}{\partial \mathbf{w}} = 0 \\ &\Rightarrow \sum_{i=0}^{m-1} 2e_i(\mathbf{x}_i) = 0 \\ &\Rightarrow \sum_{i=0}^{m-1} (\widehat{y}_i - y_i)(\mathbf{x}_i) = 0 \\ &\Rightarrow \sum_{i=0}^{m-1} (\mathbf{wx}_i - y_i)(\mathbf{x}_i) = 0 \\ &\Rightarrow \mathbf{w} \sum_{i=0}^{m-1} \mathbf{x}_i^T \mathbf{x}_i = \sum_{i=0}^{m-1} y_i \mathbf{x}_i \\ &\Rightarrow \mathbf{w} = \left[\sum_{i=0}^{m-1} \mathbf{x}_i^T \mathbf{x}_i \right]^{-1} \sum_{i=0}^{m-1} y_i \mathbf{x}_i \end{aligned} \quad (6)$$

Lo cual da como resultado los pesos \mathbf{w} que se obtienen como sigue:

$$\mathbf{w} = \left[\sum_{i=0}^{m-1} \mathbf{x}_i^T \mathbf{x}_i \right]^{-1} \sum_{i=0}^{m-1} y_i \mathbf{x}_i \quad (7)$$

2.2. Algoritmo de los mínimos cuadrados recursivos

El algoritmo de los mínimos cuadrados recursivos se usa para minimizar funciones, no solo como una función de costo o regresión no lineal. Se usan unos pasos pequeños en la dirección del mínimo de una función de costo.

Para obtener el algoritmo de los mínimos cuadrados recursivos, en primer lugar, se consideran los pesos \mathbf{w} obtenidos en (7). Se define el término G_{m-1} como sigue:

$$G_{m-1} = \sum_{i=0}^{m-1} \mathbf{x}_i^T \mathbf{x}_i \quad (8)$$

Substituyendo G_{m-1} de (8) en (7) para obtener los pesos \mathbf{w} en las iteraciones m y $m + 1$:

$$\begin{aligned} \mathbf{w}_m &= G_{m-1}^{-1} \sum_{i=0}^{m-1} y_i \mathbf{x}_i \\ \mathbf{w}_{m+1} &= G_m^{-1} \sum_{i=0}^m y_i \mathbf{x}_i \end{aligned} \quad (9)$$

Ahora, se obtiene la siguiente sumatoria:

$$\sum_{i=0}^m y_i \mathbf{x}_i = \sum_{i=0}^{m-1} y_i \mathbf{x}_i + y_m \mathbf{x}_m \quad (10)$$

Realizando operaciones matemáticas, se obtiene:

$$\begin{aligned}
\sum_{i=0}^m y_i \mathbf{x}_i &= \sum_{i=0}^{m-1} y_i \mathbf{x}_i + y_m \mathbf{x}_m \\
\Rightarrow \sum_{i=0}^m y_i \mathbf{x}_i &= G_{m-1} \mathbf{w}_m + y_m \mathbf{x}_m \\
\Rightarrow \sum_{i=0}^m y_i \mathbf{x}_i &= G_{m-1} \mathbf{w}_m + y_m \mathbf{x}_m + \mathbf{x}_m^T \mathbf{x}_m \mathbf{w}_m - \mathbf{x}_m^T \mathbf{x}_m \mathbf{w}_m \\
\Rightarrow \sum_{i=0}^m y_i \mathbf{x}_i &= (G_{m-1} + \mathbf{x}_m^T \mathbf{x}_m) \mathbf{w}_m + \mathbf{x}_m (y_m - \mathbf{w}_m^T \mathbf{x}_m) \quad (11) \\
\Rightarrow \sum_{i=0}^m y_i \mathbf{x}_i &= G_m \mathbf{w}_m - \mathbf{x}_m e_m \\
\Rightarrow G_m^{-1} \sum_{i=0}^m y_i \mathbf{x}_i &= \mathbf{w}_m - G_m^{-1} \mathbf{x}_m e_m \\
\Rightarrow \mathbf{w}_{m+1} &= \mathbf{w}_m - G_m^{-1} \mathbf{x}_m e_m
\end{aligned}$$

Entonces, el algoritmo de los mínimos cuadrados recursivos es la adaptación recursiva de los pesos \mathbf{w}_{m+1} que se describe como sigue:

$$\mathbf{w}_{m+1} = \mathbf{w}_m - P_m \mathbf{x}_m e_m \quad (12)$$

donde $P_m = G_m^{-1}$ representa la velocidad de aprendizaje la cual no es parte del modelo mismo, pero es usada para encontrar una solución, si P_m es muy grande no habrá convergencia, y si P_m es muy pequeño el algoritmo convergerá de manera lenta. \mathbf{w}_{m+1} representan los pesos (2), (7) para la iteración $m+1$, \mathbf{w}_m representan los pesos (2), (7) para la iteración m , \mathbf{x}_m representan las características (2) para la iteración m , y e_m representa el error de aprendizaje (3) para la iteración m .

Ahora de la definición (8), realizando operaciones matemáticas, se obtiene:

$$\begin{aligned}
G_m &= \sum_{i=0}^{m-1} \mathbf{x}_i^T \mathbf{x}_i + \mathbf{x}_m^T \mathbf{x}_m \\
\Rightarrow G_m &= G_{m-1} + \mathbf{x}_m^T \mathbf{x}_m \quad (13)
\end{aligned}$$

Utilizando (13), $P_m = G_m^{-1}$, y el teorema de la matriz inversa, se obtiene:

$$\begin{aligned}
G_m^{-1} &= (G_{m-1} + \mathbf{x}_m^T \mathbf{x}_m)^{-1} \\
\Rightarrow G_m^{-1} &= G_{m-1}^{-1} - \frac{1}{1 + \mathbf{x}_m^T G_{m-1}^{-1} \mathbf{x}_m} G_{m-1}^{-1} \mathbf{x}_m \mathbf{x}_m^T G_{m-1}^{-1} \quad (14) \\
\Rightarrow P_m &= P_{m-1} - \frac{1}{1 + \mathbf{x}_m^T P_{m-1} \mathbf{x}_m} P_{m-1} \mathbf{x}_m \mathbf{x}_m^T P_{m-1}
\end{aligned}$$

Por lo cual, se tiene la siguiente ecuación para actualizar la velocidad de aprendizaje:

$$P_m = P_{m-1} - \frac{1}{1 + \mathbf{x}_m^T P_{m-1} \mathbf{x}_m} P_{m-1} \mathbf{x}_m \mathbf{x}_m^T P_{m-1} \quad (15)$$

donde $P_m = G_m^{-1}$.

Tomando en consideración (2), (3), (12), y (15), el algoritmo de los mínimos cuadrados recursivos está dado por las ecuaciones:

$$\begin{aligned}
\hat{y}_m &= \mathbf{w}_m^T \mathbf{x}_m \\
e_m &= \hat{y}_m - y_m \\
P_m &= P_{m-1} - \frac{1}{1 + \mathbf{x}_m^T P_{m-1} \mathbf{x}_m} P_{m-1} \mathbf{x}_m \mathbf{x}_m^T P_{m-1} \quad (16) \\
\mathbf{w}_{m+1} &= \mathbf{w}_m - P_m \mathbf{x}_m e_m
\end{aligned}$$

donde P_m representa la velocidad de aprendizaje, e_m representa el error de aprendizaje, y_m representa la salida, \hat{y}_m representa la predicción de la salida, \mathbf{x}_m representan las características, y \mathbf{w}_m representan los pesos.

La Figura 2 muestra dos ejemplos del funcionamiento del aprendizaje de la posición de la mano del usuario mediante los mínimos cuadrados recursivos y del perfil de trayectoria. Los detalles del perfil de trayectoria se describen en la siguiente subsección.

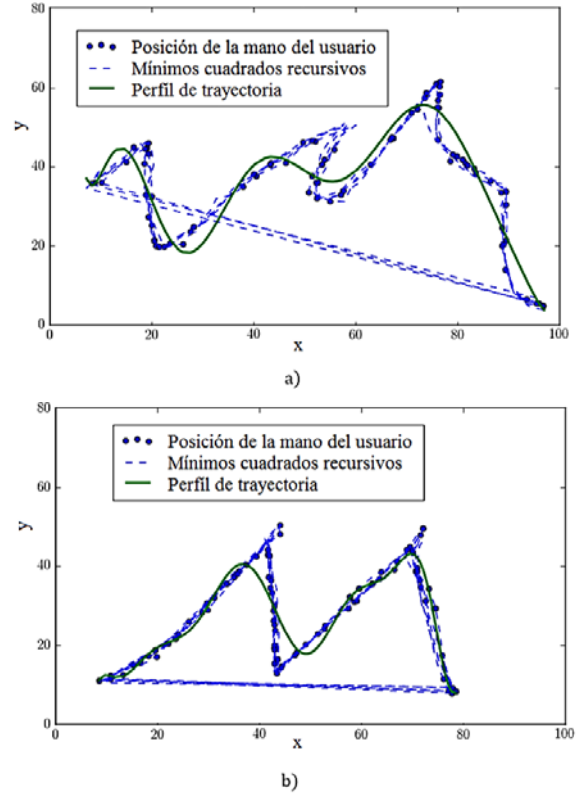


Figura 2: Dos ejemplos del aprendizaje de la posición de la mano del usuario

2.3. El perfil de la trayectoria

El manipulador debe moverse desde una posición inicial q_0 , a una final q_f , en un tiempo t_f . Esta transición debe estar regulada por leyes que contemplen el movimiento del manipulador es conocido como un perfil de trayectoria de interpoladores a tramos que permite definir la evolución del movimiento con respecto al tiempo.

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (17)$$

Si n es el número de periodos transitorios en la trayectoria, las ecuaciones que definen el movimiento del manipulador para la posición son:

$$\begin{aligned}
q(t) &= q_f \left(1 - \frac{30(n-2)}{32+30(n-2)} \right) \\
&\quad * \left[\frac{5}{4} \left(\frac{t}{t_f} \right)^3 n^3 - \frac{15}{16} \left(\frac{t}{t_f} \right)^4 n^4 + \frac{3}{16} \left(\frac{t}{t_f} \right)^5 n^5 \right] \\
&\quad \forall t, 0 \leq t \leq \frac{t_f}{n} \\
q(t) &= q \left(\frac{t_f}{n} \right) + \dot{q} \left(\frac{t_f}{n} \right) \left(t - \frac{t_f}{n} \right) \\
&\quad \forall t, \frac{t_f}{n} \leq t \leq \frac{n-1}{n} t_f \\
q(t) &= q_f \left(1 - \frac{30(n-2)}{32+30(n-2)} \right) \\
&\quad * \left[\frac{5}{4} \left(\frac{t - \frac{n-2}{n} t_f}{t_f} \right)^3 n^3 - \frac{15}{16} \left(\frac{t - \frac{n-2}{n} t_f}{t_f} \right)^4 n^4 + \frac{3}{16} \left(\frac{t - \frac{n-2}{n} t_f}{t_f} \right)^5 n^5 \right] \\
&\quad \forall t, \frac{n-1}{n} t_f \leq t \leq t_f
\end{aligned} \quad (18)$$

Las ecuaciones que definen el movimiento del manipulador para la velocidad son:

$$\begin{aligned}
 \dot{q}(t) &= q_f \left(1 - \frac{30(n-2)}{32+30(n-2)} \right) \\
 &* \left[\frac{15}{4} \frac{t^2}{t_f^3} n^3 - \frac{15}{4} \frac{t^3}{t_f^4} n^4 + \frac{15}{16} \frac{t^4}{t_f^5} n^5 \right] \\
 &\forall t, 0 \leq t \leq \frac{t_f}{n} \\
 \dot{q}(t) &= \dot{q} \left(\frac{t_f}{n} \right) \\
 &\forall t, \frac{t_f}{n} \leq t \leq \frac{n-1}{n} t_f \\
 \dot{q}(t) &= q_f \left(1 - \frac{30(n-2)}{32+30(n-2)} \right) \\
 &* \left[\frac{15}{4} \frac{(t-\frac{n-2}{n}t_f)^2}{t_f^3} n^3 - \frac{15}{4} \frac{(t-\frac{n-2}{n}t_f)^3}{t_f^4} n^4 + \frac{15}{16} \frac{(t-\frac{n-2}{n}t_f)^4}{t_f^5} n^5 \right] \\
 &\forall t, \frac{n-1}{n} t_f \leq t \leq t_f
 \end{aligned} \tag{19}$$

Las ecuaciones que definen el movimiento del manipulador para la aceleración son:

$$\begin{aligned}
 \ddot{q}(t) &= q_f \left(1 - \frac{30(n-2)}{32+30(n-2)} \right) \\
 &* \left[\frac{15}{4} \frac{t}{t_f^3} n^3 - \frac{45}{4} \frac{t^2}{t_f^4} n^4 + \frac{15}{4} \frac{t^3}{t_f^5} n^5 \right] \\
 &\forall t, 0 \leq t \leq \frac{t_f}{n} \\
 \ddot{q}(t) &= 0 \\
 &\forall t, \frac{t_f}{n} \leq t \leq \frac{n-1}{n} t_f \\
 \ddot{q}(t) &= q_f \left(1 - \frac{30(n-2)}{32+30(n-2)} \right) \\
 &* \left[\frac{15}{4} \frac{(t-\frac{n-2}{n}t_f)}{t_f^3} n^3 - \frac{45}{4} \frac{(t-\frac{n-2}{n}t_f)^2}{t_f^4} n^4 + \frac{15}{4} \frac{(t-\frac{n-2}{n}t_f)^3}{t_f^5} n^5 \right] \\
 &\forall t, \frac{n-1}{n} t_f \leq t \leq t_f
 \end{aligned} \tag{20}$$

La Figura 3 muestra las gráficas que describen los comportamientos de las ecuaciones (18), (19) y (20) para diversos valores de n donde $n=2$ está en color amarillo, $n=3$ está en color magenta, $n=4$ está en color azul, $n=5$ está en color naranja, $n=6$ está en color verde, $n=7$ está en color morado, $n=8$ está en color café, $n=9$ está en color negro y $n=10$ está en color gris.

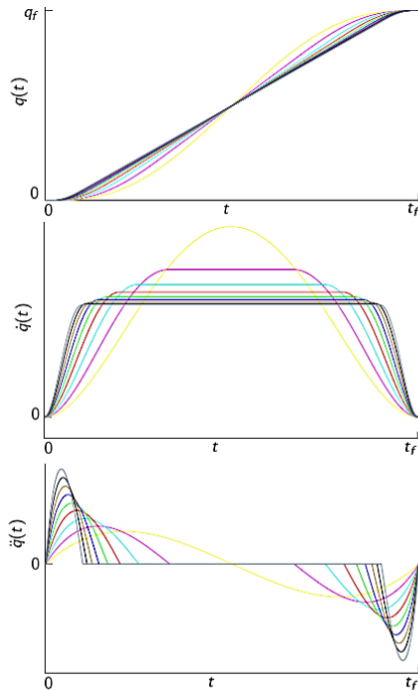


Figura 3: Comportamiento de las trayectorias para diversos valores de n

2.4. La dinámica del sistema

Los objetivos de esta sección son: a) establecer la descripción del sistema mecánico del manipulador, y obtener su modelo de cinemáticas y su modelo dinámico utilizando la formulación recursiva de Newton-Euler; b) estabilizar el sistema aplicando un controlador proporcional derivativo (PD); y c) implementar el sistema de monitoreo y control industrial embebido, utilizando la plataforma NI compactRio. Modificar, probar y validar el sistema de control en el sistema mecánico del manipulador.

La Figura 4 muestra el manipulador utilizado, que es el manipulador Amatrol Pegasus, y que posee seis grados de libertad. El estudio de la dinámica del sistema incluye el análisis de las cinemáticas directa e inversa, el modelo dinámico, y el control.



Figura 4: Manipulador Pegasus

2.5. Modelo de la cinemática directa

La Figura 5 muestra el diagrama utilizado por el método matricial de Denavit y Hartenberg Craig (2016) para establecer, de forma sistemática, un sistema de coordenadas para cada articulación.

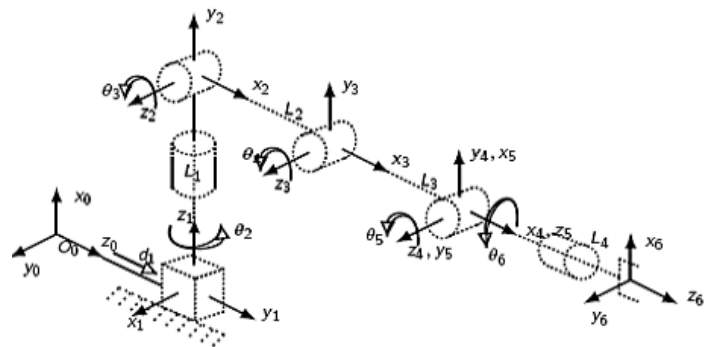


Figura 5: Cinemática directa

El resultado de la cinemática directa se describe como sigue:

$$A_{eq} = A_1 A_2 A_3 A_4 A_5 A_6$$

$$= \begin{bmatrix} c_6 & -s_{345} s_6 & & & & \\ c_2 c_{345} c_6 + s_2 s_6 & c_6 s_2 - c_2 c_{345} s_6 & & & & \\ c_{345} c_6 s_2 - c_2 s_6 & -c_2 c_6 - c_{345} s_2 s_6 & & & & \\ 0 & 0 & & & & \\ -c_{345} & L_1 + L_2 s_3 + L_3 s_{34} - L_4 c_{345} & & & & \\ c_2 s_{345} & c_2 (L_2 c_3 + L_3 c_{34} + L_4 s_{345}) & & & & \\ s_2 s_{345} & d_1 + s_2 (L_2 c_3 + L_3 c_{34} + L_4 s_{345}) & & & & \\ 0 & 1 & & & & \end{bmatrix} \quad (21)$$

donde $c_i = \cos(\theta_i)$, $c_{ij} = \cos(\theta_i + \theta_j)$, $c_{ijk} = \cos(\theta_i + \theta_j + \theta_k)$, $s_i = \sin(\theta_i)$, $s_{ij} = \sin(\theta_i + \theta_j)$, y $s_{ijk} = \sin(\theta_i + \theta_j + \theta_k)$.

2.6. Modelo de la cinemática inversa

Sea $p_j = [p_{xj}, p_{yj}, p_{zj}]$ la posición del vector medido desde el origen del j -ésimo sistema coordenado, hasta el origen del sexto sistema coordenado. Se $u_j = [u_{xj}, u_{yj}, u_{zj}]$ un vector unitario fijo respecto al eje z_6 y expresado en el j -ésimo sistema coordenado. Entonces, por definición se tiene:

$$p_6 = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$$

$$u_6 = \begin{bmatrix} 0 & 0 & 1 & 0 \end{bmatrix}^T \quad (22)$$

Cuando son expresados en el primer sistema coordenado:

$$p_1 = A_1 A_2 A_3 A_4 A_5 p_6$$

$$u_1 = A_1 A_2 A_3 A_4 A_5 u_6 \quad (23)$$

De la ecuación (23), se considera la siguiente transformación:

$$A_2^{-1} A_1^{-1} p_1 = A_3 A_4 A_5 p_6$$

$$A_2^{-1} A_1^{-1} u_1 = A_3 A_4 A_5 u_6 \quad (24)$$

De la transformación (24), se tiene el siguiente sistema de ecuaciones:

$$\begin{aligned} -L_3 s_3 s_4 + c_3 (L_2 + L_3 c_4) &= q c_2 + (r - d_1) s_2 \\ L_3 c_3 s_4 + s_3 (L_2 + L_3 c_4) &= p - L_1 \\ q s_2 - (r - d_1) c_2 &= 0 \\ c_3 (s_4 c_5 + c_4 s_5) + s_3 (c_4 c_5 - s_4 s_5) &= v c_2 + w s_2 \\ s_3 (s_4 c_5 + c_4 s_5) - c_3 (c_4 c_5 - s_4 s_5) &= u \\ -w c_2 + v s_2 &= 0 \end{aligned} \quad (25)$$

La solución del sistema de ecuaciones (25) permite encontrar los elementos de la cinemática inversa, como sigue:

$$\theta_2 = \arctan 2(w, v) = \begin{cases} v, w & \text{para } 0 \leq \frac{\pi}{2} \\ -v, w & \text{para } \frac{\pi}{2} \leq \pi \\ -v, -w & \text{para } -\pi \leq -\frac{\pi}{2} \\ v, -w & \text{para } -\frac{\pi}{2} \leq 0 \end{cases}$$

$$\theta_4 = \arctan 2 \left(\frac{d1}{\sqrt{1 - \left(\frac{qv + w(r-d_1)^2 + p - L_1^2 - L_2^2 - L_3^2}{2L_2 L_3} \right)^2}} \right) \quad (26)$$

$$\theta_3 = \arctan 2 \left(\sqrt{1 - c_3^2}, c_3 \right)$$

$$\theta_5 = \arctan 2 \left(\sqrt{1 - f_8^2}, f_8 \right)$$

$$\theta_6 = \arctan 2(l_y s_2 - l_z c_2, l_x s_{345} + c_{345} (l_y c_2 + l_z s_2))$$

donde $f_8 = \frac{f_7 (s_4 c_3 + c_4 s_3) + \sqrt{(c_4 c_3 - s_4 s_3)^2 [(s_4^2 + c_4^2)(c_3^2 + s_3^2) - f_7^2]}}{(s_4^2 + c_4^2)(c_3^2 + s_3^2)}$, $f_7 = c_3 (s_4 c_5 + c_4 s_5) + s_3 (c_4 c_5 - s_4 s_5)$.

2.7. Modelado dinámico

Se usa el método de Newton-Euler para obtener el modelo dinámico. El eje de rotación en el sistema de referencia 0 es:

$$z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (27)$$

y los ejes de rotación para las articulaciones son:

$$b_1 = (R_1^0)^T z_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$$

$$b_2 = (R_2^0)^T R_1^0 z_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$$

$$b_3 = (R_3^0)^T R_2^0 z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

$$b_4 = (R_4^0)^T R_3^0 z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$$

$$b_5 = (R_5^0)^T R_4^0 z_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T$$

$$b_6 = (R_6^0)^T R_5^0 z_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \quad (28)$$

De la formulación hacia adelante se obtienen los siguientes resultados:

$$w_0 = \alpha_0 = a_{c0} = a_{e0} = 0$$

$$w_1 = b_1 \dot{q}_1$$

$$\alpha_1 = b_1 \ddot{q}_1 + w_1 + b_1 \dot{q}_1$$

$$a_{e1} = w_1 \times r_{01} + w_1 \times (w_1 \times r_{01})$$

$$a_{c1} = w_1 \times r_{0c1} + w_1 \times (w_1 \times r_{0c1})$$

$$w_2 = (R_2^1)^T w_1 + b_2 \dot{q}_2$$

$$\alpha_2 = (R_2^1)^T \alpha_1 + b_2 \ddot{q}_2 + w_2 + b_2 \dot{q}_2 \quad (29)$$

$$a_{e2} = (R_2^1)^T a_{e1} + w_2 \times r_{12} + w_2 \times (w_2 \times r_{12})$$

$$a_{c2} = (R_2^1)^T a_{c1} + w_2 \times r_{1c2} + w_2 \times (w_2 \times r_{1c2})$$

$$w_3 = (R_3^2)^T w_2 + b_3 \dot{q}_3$$

$$\alpha_3 = (R_3^2)^T \alpha_2 + b_3 \ddot{q}_3 + w_3 + b_3 \dot{q}_3$$

$$a_{e3} = (R_3^2)^T a_{e2}$$

$$a_{c3} = (R_3^2)^T a_{c2}$$

$$w_4 = (R_4^3)^T w_3 + b_4 \dot{q}_4$$

$$\alpha_4 = (R_4^3)^T \alpha_3 + b_4 \ddot{q}_4 + w_4 + b_4 \dot{q}_4$$

$$a_{e4} = (R_4^3)^T a_{e3} + w_4 \times r_{34} + w_4 \times (w_4 \times r_{34})$$

$$a_{c4} = (R_4^3)^T a_{c3} + w_4 \times r_{3c4} + w_4 \times (w_4 \times r_{3c4})$$

$$w_5 = (R_5^4)^T w_4 + b_5 \dot{q}_5$$

$$\alpha_5 = (R_5^4)^T \alpha_4 + b_5 \ddot{q}_5 + w_5 + b_5 \dot{q}_5 \quad (30)$$

$$a_{e5} = (R_5^4)^T a_{e4}$$

$$a_{c5} = (R_5^4)^T a_{c4}$$

$$w_6 = (R_6^5)^T w_5 + b_6 \dot{q}_6$$

$$\alpha_6 = (R_6^5)^T \alpha_5 + b_6 \ddot{q}_6 + w_6 + b_6 \dot{q}_6$$

$$a_{e6} = (R_6^5)^T a_{e5} + w_6 \times r_{56} + w_6 \times (w_6 \times r_{56})$$

$$a_{c6} = (R_6^5)^T a_{c5} + w_6 \times r_{5c6} + w_6 \times (w_6 \times r_{5c6})$$

De la formulación hacia atrás se obtienen los siguientes resultados:

$$F_7 = \tau_7 = 0$$

$$g_0 = \begin{bmatrix} 0 & 0 & -g \end{bmatrix}^T$$

$$g_6 = (R_6^0)^T g_0$$

$$F_6 = m_6 a_{c6} - m_6 g_6$$

$$\tau_6 = -F_6 \times r_{5c6} + w_6 \times (l_6 w_6) + l_6 \alpha_6$$

$$g_5 = (R_5^0)^T g_0$$

$$F_5 = R_6^5 F_6$$

$$\tau_5 = R_6^5 \tau_6 + w_5 \times (l_5 w_5) + l_5 \alpha_5$$

$$g_4 = (R_4^0)^T g_0$$

$$F_4 = R_5^4 F_5 + m_4 a_{c4} - m_4 g_4$$

$$\tau_5 = R_5^4 \tau_5 - F_4 \times r_{3c4} + R_5^4 F_5 \times r_{4c4} + w_4 \times (l_4 w_4) + l_4 \alpha_4 \quad (31)$$

$$\begin{aligned}
 g_3 &= (R_3^0)^T g_0 \\
 F_3 &= R_4^3 F_4 \\
 \tau_3 &= R_4^3 \tau_4 + w_3 \times (l_3 w_4) + l_3 \alpha_3 \\
 g_2 &= (R_2^0)^T g_0 \\
 F_2 &= R_3^2 F_3 + m_2 a_{c2} - m_2 g_2 \\
 \tau_2 &= R_3^2 \tau_3 - F_2 \times r_{1c2} + R_3^2 F_3 \times r_{2c2} + w_2 \times (l_2 w_2) + l_2 \alpha_2 \\
 g_1 &= (R_1^0)^T g_0 \\
 F_1 &= R_2^1 F_2 + m_1 a_{c1} - m_1 g_1 \\
 \tau_1 &= R_2^1 \tau_2 - F_1 \times r_{0c1} + R_2^1 F_2 \times r_{1c1} + w_1 \times (l_1 w_1) + l_1 \alpha_1
 \end{aligned} \tag{32}$$

El modelo dinámico del sistema de las ecuaciones (31) y (32) se puede reescribir como sigue:

$$\begin{aligned}
 M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) &= u \\
 q &= \begin{bmatrix} q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{bmatrix}^T \\
 u &= \begin{bmatrix} \tau_1 & \tau_2 & \tau_3 & \tau_4 & \tau_5 & \tau_6 \end{bmatrix}^T
 \end{aligned} \tag{33}$$

donde $M(q) \in \mathbb{R}^{6 \times 6}$, $C(q, \dot{q}) \in \mathbb{R}^{6 \times 6}$, y $G(q) \in \mathbb{R}^{6 \times 1}$.

2.8. El control del sistema

El control proporcional derivativo (PD) ha demostrado trabajar de manera óptima con los sistemas industriales. El controlador se aplica independientemente a cada una de las articulaciones del manipulador, lo que quiere decir que cada articulación es controlada como una sola entrada y el controlador otorga una sola salida. El control PD para cada articulación es:

$$u_i = -K_{pi}(p_{refi} - q_i) - K_{di}\dot{q}_i \tag{34}$$

donde $i = 1, \dots, 6$, p_{refi} es la referencia deseada para la articulación i , $K_{pi} \in \mathbb{R}$ es la ganancia proporcional, y K_{di} es la ganancia derivativa.

3. Resultados experimentales

Los resultados experimentales se dividen en dos partes: el aprendizaje automático y el controlador.

3.1. Resultados del aprendizaje automático

El sensor Kinect for Windows es utilizado principalmente como interfaz natural de usuario para videojuego. Este dispositivo puede hacer seguimiento en tres dimensiones del cuerpo humano usando su cámara de profundidad y su cámara RGB. Éstas permiten, en combinación con algoritmos de aprendizaje automático, obtener la posición de 20 elementos del cuerpo humano a una frecuencia de 30 Hz, con una resolución de 640x480 píxeles. La cámara de profundidad basa su funcionamiento en la proyección y recepción de patrones de luces láser infra-rojas. La Figura 6 muestra la colocación del sensor para la adquisición de datos. El sensor Kinect for Windows es conectado a un puerto USB 2.0 de una computadora con procesador AMD A8-7600 Radeon R7 con 10 núcleos, velocidad de procesamiento de 3.10 GHz, 8 GB de memoria RAM y sistema operativo Windows 7 de 64 bits. Los datos del sensor son extraídos mediante el software Matlab R2014a con las herramientas de Image acquisition toolbox for Kinect sensor y Kinect for Windows SDK.



Figura 6: Colocación del sensor

La Figura 7 muestra la interfaz de usuario que se escribió en Matlab usando las librerías Image acquisition toolbox y Kinect for Windows SDK. La interfaz de usuario, contiene un botón llamado Info que permite conocer el estado del sensor. Éste muestra la resolución de ambas cámaras al momento de ser pulsado. Así, también se muestran dos listas desplegables, la primera, llamada Select hand permite configurar la mano a seguir de entre Right hand y Left hand. La segunda lista desplegable llamada Pose permite configurar el modo de seguimiento de la persona de entre sentado Seat o parado Standing. Al presionar el botón Start Video se inicializa el sensor configurado con las posiciones previamente descritas. Posteriormente, se adquiere la imagen del sensor de profundidad y se muestra. A partir de este punto el programa busca si existe el movimiento de alguna persona en la escena.



Figura 7: Interfaz de usuario para la captura de puntos

De acuerdo a la información de los píxeles se crean centros de masa en tres dimensiones a través de un estimador de densidad para cada parte del cuerpo con respecto a la superficie de cada una de estas. El estimador usa como información tres diferentes vistas: de frente, de lado, y superior las cuales son también estimadas. Se generan 20 articulaciones del cuerpo humano con tres estados para cada una de estas: encontrado, no encontrado e inferido. Después de la obtención de las articulaciones, se dibuja una serie de líneas que representan el esqueleto y se coloca un marcador sobre la mano que se está siguiendo. Mientras el botón Stop Video no sea presionado, los datos de la posición de la mano se guardan en una lista junto con el tiempo en que fueron tomados. Si cierra la interfaz de usuario, los datos se guardan en un archivo para su procesamiento. La Figura 8 muestra el diagrama de flujo de la interfaz de usuario para la captura de datos.

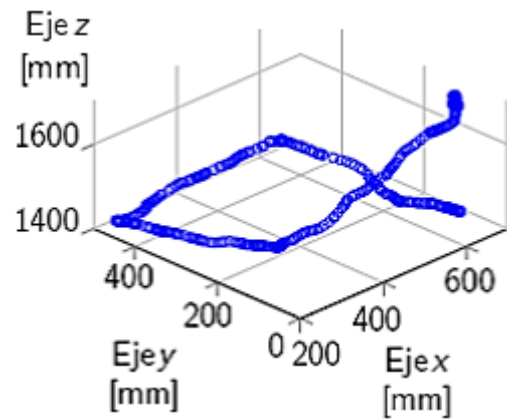


Figura 9: Puntos adquiridos por el sistema

Tabla 1: Adquisición de datos

Tiempo	Posición x	Posición y	Posición z
0,0000	523,90873	345,24545	1345,11650
0,07157	525,29728	344,84822	1343,32275
0,13678	527,31872	345,91395	1341,09604
0,20471	528,66966	346,75002	1339,42235

Como el problema del aprendizaje de los movimientos de la mano no puede ser completado por un programa que no se modifique a sí mismo, ya que no sería factible escribir miles de programas, uno para cada patrón de movimiento de la mano del usuario, es necesario usar otro método como el de dar a la computadora la habilidad de aprender. El algoritmo que se utiliza como prueba en este estudio, es el de mínimos cuadrados recursivos, el cual, cumple con la característica de modificarse a sí mismo.

3.2. Resultados del controlador

La Figura 10 muestra el diagrama del control industrial simplificado del manipulador. El módulo de adquisición de datos se compone de un sensor Kinect for Windows conectado a una computadora Dell Optitex 780 mediante el puerto USB 2.0, el sensor transmite imágenes de profundidad a la computadora para que éstas sean analizadas por su software de adquisición de datos. La pantalla de la computadora muestra una interfaz de usuario, como se aprecia en la Figura 7 que permite adquirir los movimientos de la mano del usuario para que estos sean trabajados por el algoritmo de aprendizaje automático con el fin de generar una trayectoria aprendida que es guardada como un vector de coordenadas en un archivo de texto. El controlador integrado en tiempo real NI cRIO-9074 lee el archivo de texto generado por la computadora a través de una variable local encontrada en la red Ethernet que conecta la computadora y el controlador. El software de control, hospedado en la memoria FPGA del controlador manda las trayectorias de movimiento al módulo de control de servomotores de corriente directa (DC) a través de los seis controladores de motor NI 9505, que son del tipo proporcional derivativo y que utilizan como variable controlada la información de las posiciones de las articulaciones otorgadas por el módulo de entradas y salidas digitales NI 9375 en razón de la conexión de los encoders incrementales conectados a dicho módulo y que permite mover las articulaciones

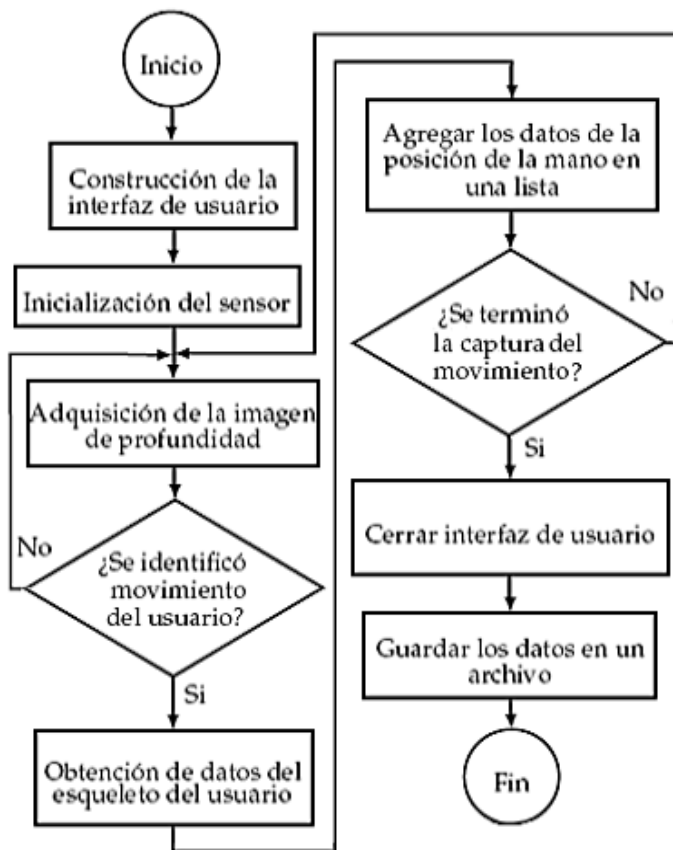


Figura 8: Diagrama de flujo de la captura de datos

De acuerdo a las especificaciones técnicas del sistema de cómputo utilizado se destaca que por cada segundo de captura el sistema registra 13 capturas de datos analizando 30 imágenes de profundidad, es decir, en total cada captura de la posición de la mano tarda 77 ms en realizarse. La Figura 9 muestra la gráfica de los datos adquiridos del movimiento de la mano durante el primer segundo, y la Tabla 1 muestra la adquisición de datos durante el primer segundo donde el tiempo es en s y las posiciones son en mm.

del manipulador. Los módulos mencionados funcionan gracias a la alimentación de energía eléctrica otorgada por el módulo de potencia eléctrica HTS60F24 y NI PS15.

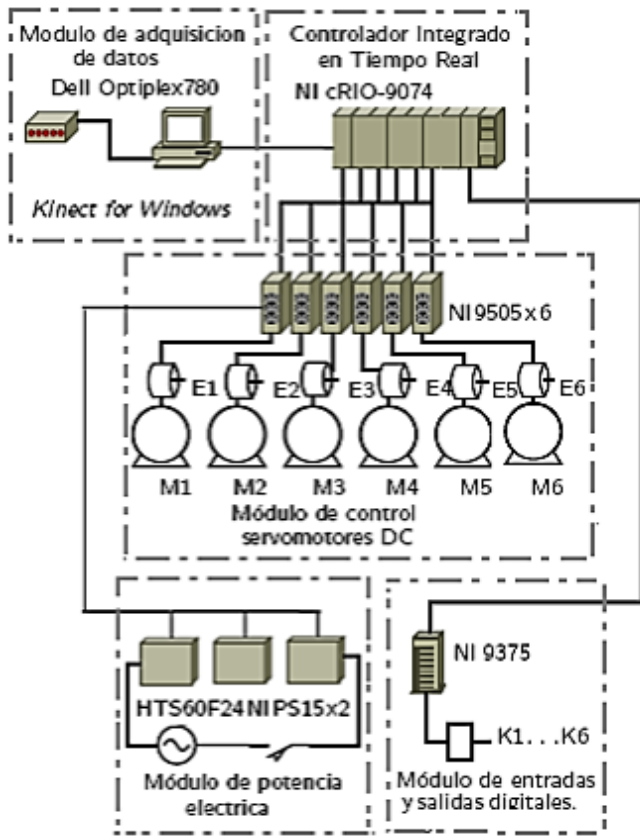


Figura 10: Diagrama del control industrial del manipulador

La Figura 11 muestra el manipulador usado en los experimentos, el desplazamiento 1 es lineal, mientras que los desplazamientos 2, 3, 4, 5, y 6 son rotacionales.



Figura 11: El manipulador

La Figura 12 muestra la instrumentación del controlador usado en los experimentos que corresponde a los elementos descritos en la Figura 10. La Figura 12a) muestra en módulo de entradas y salidas digitales NI 9375. La Figura 12b) en el lado izquierdo muestra en controlador integrado en tiempo real NI cRIO-9074, y en el lado derecho muestra el dispositivo HTS60F24. La Figura 12c) en el lado izquierdo muestra el módulo de potencia NI PS15x2, y en el lado derecho muestra el módulo de control de los motores de corriente directa (DC) NI 9505x6, donde x2 corresponde a que son dos dispositivos iguales y x6 corresponde a que son seis dispositivos iguales.

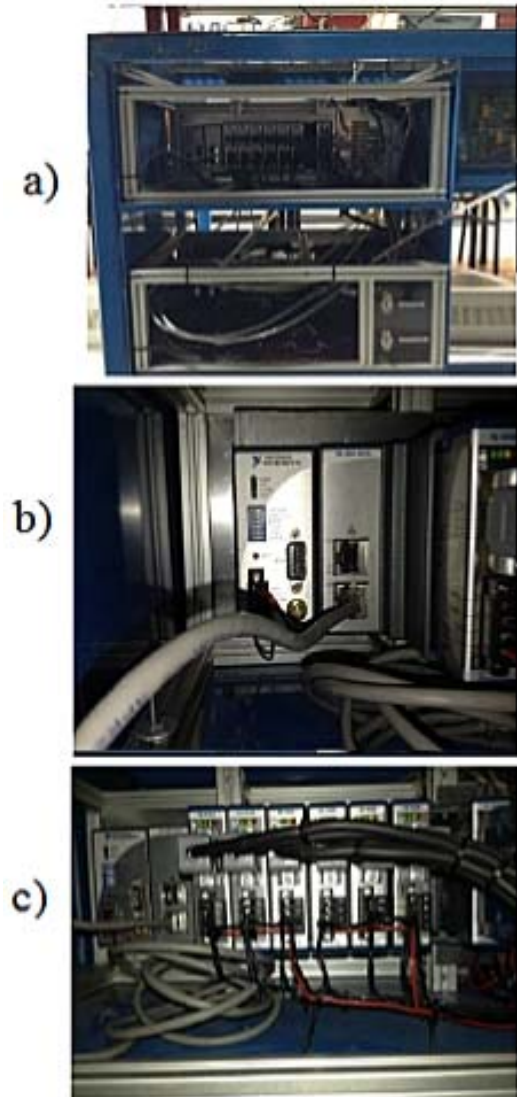


Figura 12: La instrumentación del controlador

De acuerdo a la captura de los movimientos del manipulador, se puede establecer una comparación entre cada uno de los movimientos de las articulaciones del manipulador reales y los movimientos planeados teóricos, en posición, velocidad y aceleración. Las Figuras 13, 14, 15, 16, 17, y 18 muestran los resultados del control propuesto para las articulaciones 1, 2, 3, 4, 5, y 6, respectivamente.

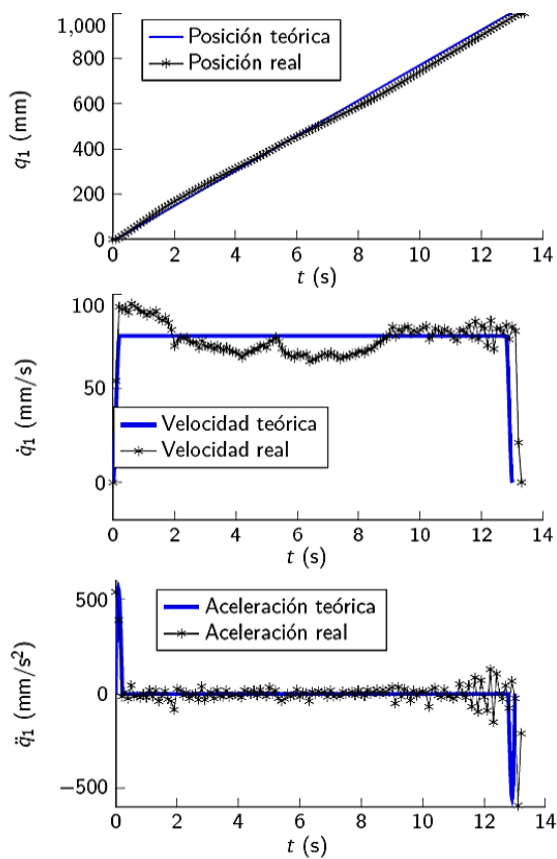


Figura 13: Resultados para la articulación 1

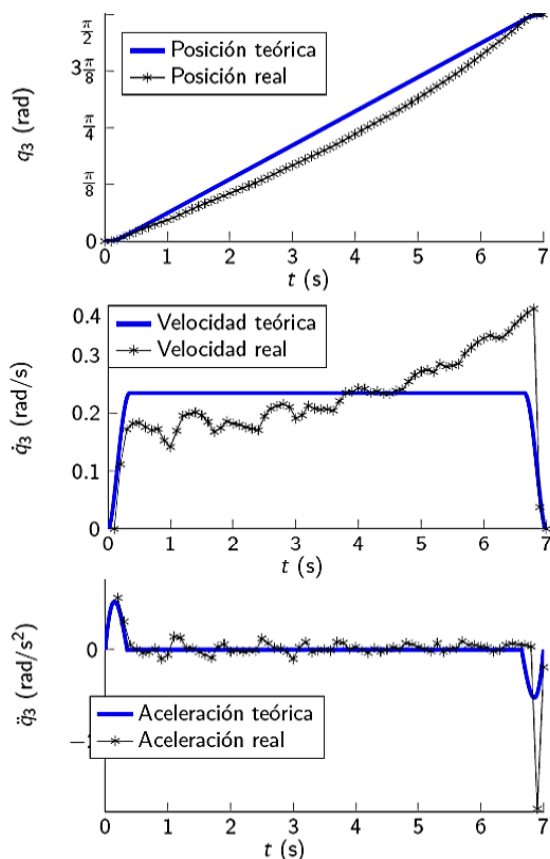


Figura 15: Resultados para la articulación 3

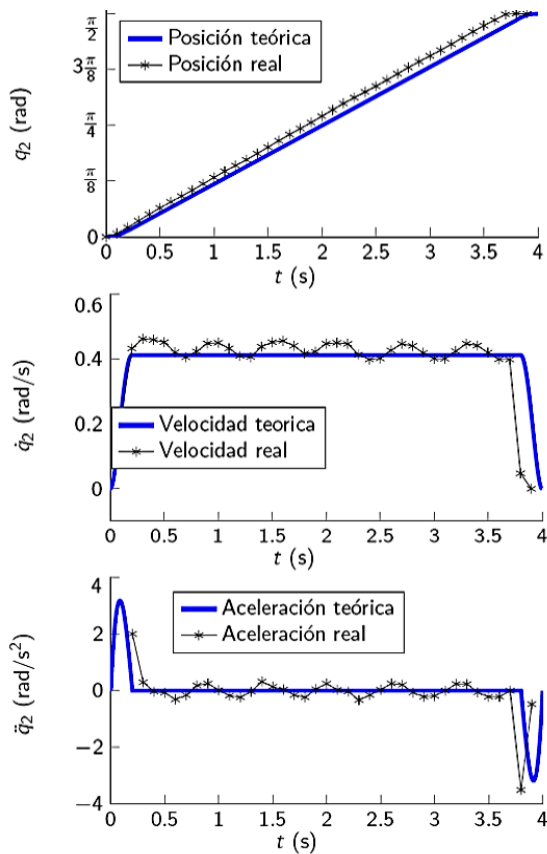


Figura 14: Resultados para la articulación 2

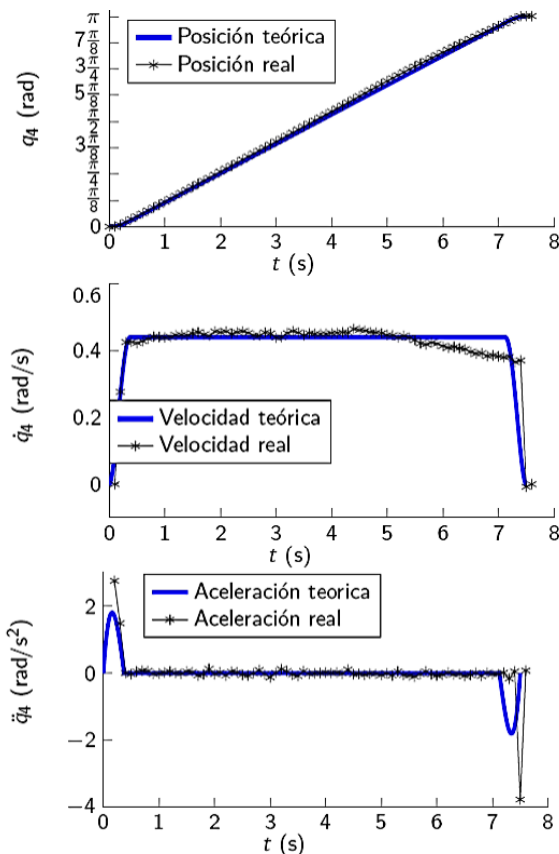


Figura 16: Resultados para la articulación 4

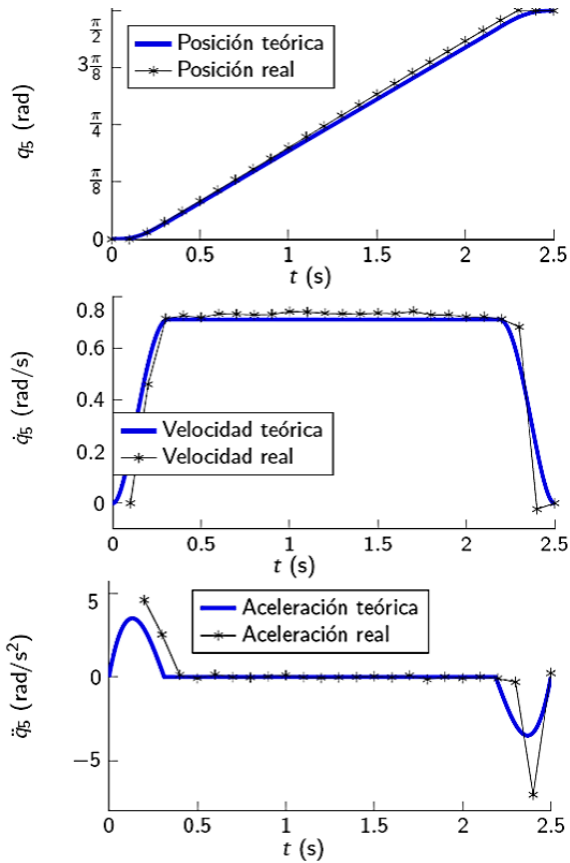


Figura 17: Resultados para la articulación 5

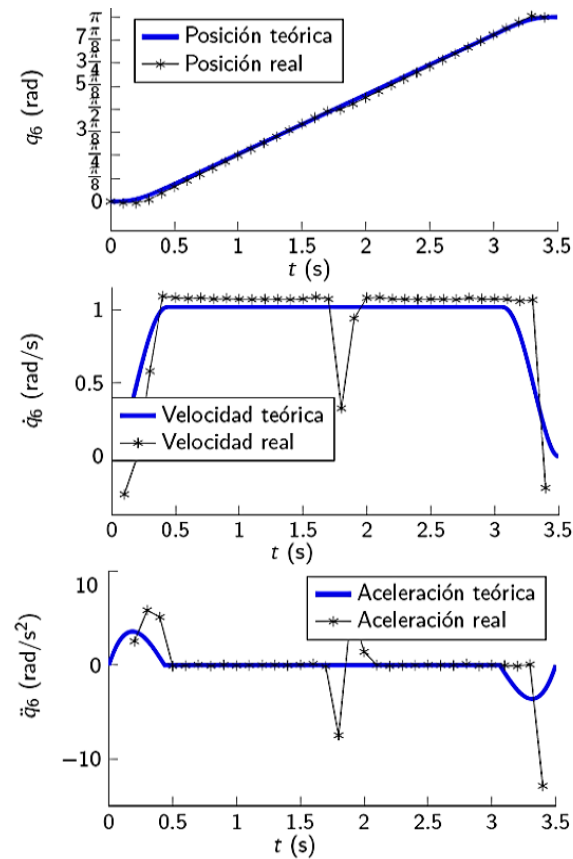


Figura 18: Resultados para la articulación 6

Las posiciones, velocidades, y aceleraciones teóricas y experimentales del manipulador son muy parecidas, pero no son completamente iguales, esto puede deberse a que algunas variables no fueron tomadas en cuenta por parte del modelo dinámico. Sin embargo, estas posiciones, velocidades y aceleraciones teóricas y experimentales tienen una gran similitud. Así también, se ha demostrado la eficiencia de la captura de datos bajo condiciones controladas. El usar el algoritmo de mínimos cuadrados recursivos, permitió aprender y generar la trayectoria de manera satisfactoria. El usar un perfil de trayectoria de interpoladores a tramos permitió suavizar el movimiento del manipulador cuando se presentó un cambio de sentido en alguna de las articulaciones. Por lo descrito anteriormente, se determina que los métodos funcionan adecuadamente.

4. Conclusiones

El objetivo principal de este trabajo es el aprendizaje automático de los movimientos de la mano de un usuario por parte de un manipulador, a partir de la enseñanza de los mismos, con la finalidad de que no sea necesario programarlos explícitamente, y aún así tengan un uso real en la industria. El algoritmo de aprendizaje de los mínimos cuadrados recursivos, cumple su función, el aprendizaje y la generación de la trayectoria de la mano del usuario a partir de los datos otorgados por el sensor. Se ha demostrado que el cálculo de la dinámica de un manipulador no es tarea fácil y que además su implementación en un sistema real es básicamente un esfuerzo de prueba y error. A pesar de esto, el modelo dinámico fue llevado directamente a una plataforma de desarrollo embebida industrial de manera eficiente, lo que permite a futuro, poder llevar el mismo sistema a cualquier manipulador industrial en caso de que se requiera. El modelo dinámico es limitado a condiciones ideales de funcionamiento, las cuales pueden perderse en cualquier momento, por ejemplo en un choque, por lo que el modelo dinámico del manipulador tiene un amplio potencial de mejora al incluir cálculos más exactos para los centros de masa y las matrices de inercia, así como la inclusión de modelos que incorporen la fricción en las articulaciones. El uso de la segunda versión del sensor de profundidad aumenta considerablemente la adquisición de datos, sobre todo en el eje z. El diseño de los algoritmos de aprendizaje es necesario para aprender los comportamientos generados por la mano de un usuario para generar la trayectoria. El diseño de los algoritmos de perfil de trayectoria es necesario para incluir la evasión de los obstáculos que puedan presentarse en la trayectoria generada.

Agradecimientos

Los autores agradecen al editor y a los revisores por sus valiosos comentarios y sugerencias que permitieron mejorar esta investigación significativamente. Así como al Instituto Politécnico Nacional, al Consejo Nacional de Ciencia y Tecnología (CONACYT), y a la Secretaría de Investigación y Posgrado (SIP) por el apoyo otorgado.

Referencias

- Calderón C. A., Ramírez C., Barros V., Punin G., 2017. Design and deployment of grasp control system applied to robotic hand prosthesis, *IEEE Latin America Transactions*, 15, (2), 181-188.
- Candelas F. A., Torres F., Gil P., Ortiz F., Puente S., Pomares J., 2004. Laboratorio virtual remoto para robótica y evaluación de su impacto en la docencia, *Revista Iberoamericana de Automática e Informática Industrial*, 1, (2), 49-57.
- Craig J. J., 2006. *Introducción a la robótica*, tercera edición, New Jersey, Prentice Hall.
- Espinosa F., Lázaro J. L., Olivares J., 2018. Proyecto alcor: contribuciones a la optimización del guiado remoto de robots en espacios inteligentes, *Revista Iberoamericana de Automática e Informática industrial*. DOI: 10.4995/riai.2018.9199
- García M. A., Gallardo J., Rodríguez R., Alcaraz L. A., 2017. A new four-degrees-of-freedom parallel manipulator, *IEEE Latin America Transactions*, 15, (5), 928-934.
- Hernández K., Bacca B., Posso B., 2017. Multi-goal path planning autonomous system for picking up and delivery tasks in mobile robotics, *IEEE Latin America Transactions*, 15, (2), 232-238.
- Hernández Guzmán V. M., Antonio Cruz M., Silva Ortigoza R., 2016. Linear state feedback regulation of a furuta pendulum: design based on differential flatness and root locus, *IEEE Access*, 4, 8721-8736.
- Kanj S., Abdallah F., 2016. Editing training data for multi-label classification with the k-nearest neighbor rule, *Pattern Analysis and Applications*, 19, (1), 145-161.
- Li X., Pan Y., Chen G., Yu H., 2017. Multi-modal control scheme for rehabilitation robotic exoskeletons, *International Journal of Robotics Research*, 36, (5), 759-777.
- Liu Z., Liu J., 2017. Adaptive iterative learning boundary control of a flexible manipulator with guaranteed transient performance, *Asian Journal of Control*, 19, (4), 1-12.
- Marín L., Valles M., Soriano A., Valera A., Albertos P., 2014. Event-based localization in ackermann steering limited resource mobile robots, *IEEE/ASME Transactions on Mechatronics*, 19, (4), 1171-1182.
- Marín L., Valles M., Soriano A., Valera A., Albertos P., 2013. Multi sensor fusion framework for indoor-outdoor localization of limited resource mobile robots, *Sensors*, 13, 14133-14160.
- Martín A., Terrile S., Barrientos A., del Cerro J., 2018. Robots hiper-redundantes: clasificación, estado del arte y problemática, *Revista Iberoamericana de Automática e Informática Industrial*. DOI: 10.4995/riai.2018.9207
- Méndez Perez J. A., Torres S., Rebozo J. A., Rebozo H., 2011. Estrategias de control en la práctica de anestesia, *Revista Iberoamericana de Automática e Informática industrial*, 8, 241-249.
- Ngoc Thanh H. L. N., Kyung Hong S., 2018. Completion of collision avoidance control algorithm for multicopters based on geometrical constraints, *IEEE Access*, 6, 27111-27126.
- Olivares M., Albertos P., 2014. Linear control of the flywheel inverted pendulum, *ISA Transactions*, 53, 1396-1403.
- Ortigoza R. S., Sánchez J. R. G., Guzmán V. M. H., Sánchez C. M., Aranda M. M., 2016. Trajectory tracking control for a differential drive wheeled mobile robot considering the dynamics related to the actuators and power stage, *IEEE Latin America Transactions*, 14, (2), 657-664.
- Páramo Carranza L. A., Meda Campaña J. A., Rubio J. J., Tapia Herrera R., Curtidor López A. V., Grande Meza A., Cazares Ramirez I., 2017. Discrete-time Kalman filter for Takagi-Sugeno fuzzy models, *Evolving Systems*, 8, (3) 211-219.
- Peng Y., Liu J., He W., 2017. Boundary control for a flexible inverted pendulum system based on a pde model, *Asian Journal of Control*, 19, (2), 1-10.
- Pomares J., García G. J., Peréa I., Jara C., Torres F., 2012. Cámara virtual de referencia: una aproximación para el guiado combinado de robots manipuladores mediante una única cámara, *Revista Iberoamericana de Automática e Informática Industrial*, 9, 314-323.
- Pan Y., Zhang J., Yu H., 2016. Model reference composite learning control without persistency of excitation, *IET Control Theory & Applications*, 10, (16), 1963-1971.
- Rosado W. M. A., Valdés L. G. V., Ortega A. B., Ascencio J. R., Beltrán C. D. G., 2017. Passive rehabilitation exercises with an ankle rehabilitation prototype based in a robot parallel structure, *IEEE Latin America Transactions*, 15, (1), 48-56.
- Rubio J. J., 2018. Discrete time control based in neural networks for pendulums, *Applied Soft Computing*, 68, 821-832.
- Rubio J. J., Vázquez D. M., Mujica-Vargas D., 2013. Acquisition system an approximation of brain signals. *IET Science, Measurement, and Technology*, 7, (4), 232-239.
- Rubio J. J., 2017. Interpolation neural network model of a manufactured wind turbine, *Neural Computing and Applications*, 28, (8), 2017-2028.
- Sa S. T. L., Fernandes C. C., Yanaguibashi E. A., Barros R. P., Burlamaqui A. M. F., Goncalves L. M. G., 2017. Educaval: towards assessment of educational robotics softwares, *IEEE Latin America Transactions*, 15, (4), 720-728.
- Serrano M. E., Godoy S. A., Romoli S., Scaglia G. J. E., 2018. A numerical approximation-based controller for mobile robots with velocity limitation, *Asian Journal of Control*, 20, (1), 1-13.
- Serrano F. E., Rodríguez-Gómez B. A., Cardona M., Obtención de un modelo dinámico para un robot 3RRR basado en teoría de screws, *Revista Iberoamericana de Automática e Informática Industrial*. DOI: 10.4995/riai.2018.8725
- Sun T., Zhang J., Pan Y., 2017. Active disturbance rejection control of surface vessels using composite error updated extended state observer, *Asian Journal of Control*, 19, (5), 1802-1811.
- Torres S., Méndez J. A., 2009. Seguimiento de trayectorias en robots manipuladores: revisión de soluciones y nuevas propuestas, *Revista Iberoamericana de Automática e Informática Industrial*, 6, (4), 80-92.
- Vázquez D. M., Rubio J. J., Pacheco J., 2012. Characterization framework for epileptics signals, *IET Image Processing*, 6, (9), 1227-1235.
- Zhang C., Sun T., Pan Y., 2014. Neural network observer-based finite-time formation control of mobile robots, *Mathematical Problems in Engineering*, 2014, 1-9.