



(51) Clasificación Internacional de Patentes:
G09B 25/02 (2006.01) *B25J 9/00* (2006.01)

(21) Número de la solicitud internacional:
PCT/CL2015/050056

(22) Fecha de presentación internacional:
21 de diciembre de 2015 (21.12.2015)

(25) Idioma de presentación: español

(26) Idioma de publicación: español

(30) Datos relativos a la prioridad:
3551 -2014
29 de diciembre de 2014 (29.12.2014) CL

(71) Solicitante: **UNIVERSIDAD DE SANTIAGO DE CHILE** [CL/CL]; Av. Libertador Bernardo O'Higgins 3363, Estación Central, Santiago (CL).

(72) Inventores: **URREA O., Claudio**; Av. Libertador Bernardo O'Higgins 3363, Estación Central, Santiago (CL). **KERN MOLINA, John**; Av. Libertador Bernardo O'Higgins 3363, Estación Central, Santiago (CL).

(74) Mandatario: **JOHANSSON & LANGLOIS**; San Pió X 2460, Piso 1, Providencia, Santiago (CL).

(81) Estados designados (*a menos que se indique otra cosa, para toda clase de protección nacional admisible*): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Estados designados (*a menos que se indique otra cosa, para toda clase de protección regional admisible*): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), euroasiática (AM, AZ, BY, KG, KZ, RU, TJ, TM), europea (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,

[Continúa en la página siguiente]

(54) Title: METHOD FOR DESIGNING AND IMPLEMENTING A GRAPHIC SIMULATOR FOR A REDUNDANT INDUSTRIAL ROBOT

(54) Título : MÉTODO DE DISEÑO E IMPLEMENTACIÓN DE SIMULADOR GRÁFICO PARA ROBOT INDUSTRIAL REDUNDANTE

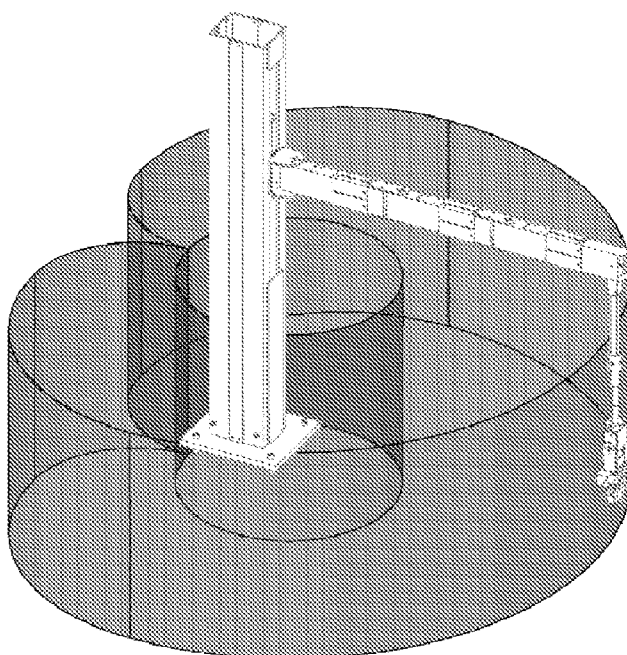


FIG. 13

(57) Abstract: The invention relates to a method for designing and implementing a graphic simulator for redundant industrial robots of N degrees of freedom (GDL), which involves the use of software tools, such as MatLab/Simulink developed by MathWorks, and MathType developed by Design Science, providing the kinematic and dynamic models of redundant industrial robots.

(57) Resumen: La presente invención describe un método de diseño e implementación de un simulador gráfico para robots Industriales redundantes de N Grados De Libertad (GDL) que contempla la utilización de herramientas de software, como por ejemplo: MatLab/Simulink, desarrollada por MathWorks, y MathType, desarrollada por Design Science, proporcionando los modelos cinemático y dinámico de robots industriales redundantes.





RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG,
CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD,
TG).

Publicada:

— con informe de búsqueda internacional (Art. 21(3))

**MÉTODO DE DISEÑO E IMPLEMENTACIÓN DE SIMULADOR GRÁFICO PARA
ROBOT INDUSTRIAL REDUNDANTE**

MEMORIA DESCRIPTIVA

Campo de aplicación

5 En la actualidad, la necesidad de ensayar modelos de simulación que se desarrollan para representar sistemas de diversa naturaleza, tales como robots industriales, se ha incrementado vertiginosamente. Sin embargo, las herramientas de software disponibles en el mercado para la realización de tales ensayos son de propósito general, y las aplicaciones desarrolladas para ensayos de manipuladores
10 redundantes específicos no están optimizadas para la resolución de modelos de robots industriales redundantes, lo que se traduce en procedimientos laboriosos de diseño, lentitud en la ejecución y falta de precisión en los resultados.

La presente invención describe un método de diseño e implementación de un simulador gráfico para robots industriales redundantes de N Grados de Libertad
15 (GDL) que contempla la utilización de herramientas de software, como por ejemplo: MatLab/Simulink, desarrollada por MathWorks, y MathType, desarrollada por Design Science, logrando modelos cinemático y dinámico de robots industriales redundantes.

Estado de la técnica

20 Existen en el mercado diversos simuladores aplicados a sistemas robotizados que presentan excelentes características de animación tridimensional, como por ejemplo:

RoboWorks (desarrollado por Newtonium) o ROBOGUIDE (desarrollado por FANUC Robotics), sin embargo, estos simuladores no pueden modificarse ya que su código fuente no está disponible para el usuario.

5 A continuación, se presentan algunas herramientas de software y trabajos realizados, que utilizan como base el programa MatLab/Simulink, que permiten la simulación de un robot industrial y donde además, su código está disponible para el usuario.

La herramientas de software: HEMERO (ref. 1), SpaceLib (ref. 2) y Robotics Toolbox (ref. 3) corresponden, cada una de ellas, a conjuntos de funciones matemáticas para
10 instalar en MatLab, denominadas normalmente Toolbox.

Éstas herramientas permiten realizar cálculos cinemáticos y dinámicos, y simular robots con diferentes estructuras. Sin embargo, todas ellas son de propósito general y, por lo tanto, no están optimizadas para simular un robot industrial redundante lo que trae como consecuencia: el requerimiento de mayor tiempo de procesamiento.

15 Los trabajos denominados: NXT SCARA (ref. 4), Model Based Robotic Control Design (ref. 5) y SCARA with Path trajectory (ref. 6) son herramientas para simulación de modelos específicos de robots, sin embargo, ninguno de ellos, está aplicado a un robot industrial con redundancia rotacional y prismática. Además, no permiten el cálculo simbólico de la cinemática y la dinámica del robot.

20 Cabe mencionar que ninguna de las herramientas señalas anteriormente establece una metodología para desarrollar un simulador gráfico para un robot redundante ni

proporcionan los modelos: cinemático y dinámico, en formato de expresión simbólica en colores.

Resumen

La presente invención describe un método de diseño e implementación de un simulador gráfico para robots industriales redundantes de N Grados de Libertad (GDL) que contempla la utilización de herramientas de software, como por ejemplo: MatLab/Simulink, desarrollada por MathWorks, y MathType, desarrollada por Design Science, logrando modelos cinemático y dinámico de robots industriales redundantes.

10 **Breve descripción de las figuras**

Figura 1: Diagrama de flujo con secuencia de etapas del método de la presente invención.

Figura 2: Esquema de un manipulador robotizado redundante de 5 GDL

Figura 3: Sistema de coordenadas de Denavit y Hartenberg Estándar considerando
15 la ubicación de los centroides para 5 GDL

Figura 4: Extracto de la función MatLab "datos_dinJk(n)"

Figura 5: Extracto de la función MatLab "dinamicajk()"

Figura 6: Extracto de la función MatLab "modJk()"

Figura 7: Extracto de la función MatLab "latexjk(in)"

20 Figura 8: Extracto de la función MatLab "simpjk(in)"

Figura 9: Extracto de la función MatLab "convjk(in)"

Figura 10: Extracto del *Script* MatLab para calcular la cinemática del robot redundante

Figura 11: Extracto del *Script* MatLab para calcular la dinámica del robot redundante

5 Figura 12: Espacio de trabajo proyectado en el plano xy

Figura 13: Vista isométrica que indica el espacio de trabajo del manipulador redundante

Figura 14: Diagrama Simulink que señala subsistemas Simulink correspondiente bloque "Brazo"

10 Figura 15: Diagrama Simulink que señala un subsistema correspondiente al interior del bloque " $[M^L - 1]$ "

Figura 16: Extracto del *Script* MatLab correspondiente a la función embebida "Embedded MATLAB Function1 " para calcular la matriz de inercia del bloque " $[M^L - 1]$ "

15 Figura 17: Diagrama Simulink que señala un subsistema que contiene un bloque "Constantes" y un bloque "Robot - 5GDL"

Figura 18: Diagrama Simulink que señala una máscara correspondiente al bloque "Brazo" del robot redundante

Figura 19: Imagen que muestra la forma de ingresar los parámetros a la máscara correspondiente al bloque "Brazo" del robot redundante

Figura 20: Diagrama Simulink que señala un subsistema correspondiente al interior de la máscara "Actuadores"

Figura 21: Extracto del *Script* MatLab correspondiente a la función embebida "Embedded MATLAB Function" para calcular la dinámica de los actuadores

5 Figura 22: Diagrama Simulink que señala una máscara correspondiente al bloque "Actuadores" del robot redundante

Figura 23: Imagen que muestra la forma de ingresar los parámetros a la máscara correspondiente al bloque "Actuadores" del robot redundante

Figura 24: Diagrama Simulink que señala la estructura del controlador "Par Cale -
10 Neuronal"

Figura 25: Diagrama Simulink que señala un subsistema del controlador "Par Cale - Neuronal"

Figura 26: Extracto del *Script* MatLab correspondiente a la función embebida "Embedded MATLAB Functionl " para calcular el vector de fuerzas centrífugas y de
15 Coriolis del bloque "Controlador"

Figura 27: Diagrama Simulink del subsistema que contiene un bloque controlador "Par Cale - Neuronal"

Figura 28: Diagrama Simulink que señala una máscara correspondiente al bloque "Controlador"

Figura 29: Diagrama Simulink que señala un subsistema que contiene bloques de trayectorias articulares pertenecientes al bloque "Trayectorias"

Figura 30: Diagrama Simulink que señala un el interior de bloque "Trayectoria2"

Figura 31: Diagrama Simulink que señala una máscara correspondiente al bloque
5 "Trayectoria" del robot redundante

Figura 32: Imagen que muestra la forma de ingresar los parámetros a la máscara correspondiente al bloque "Trayectoria" del robot redundante

Figura 33: Aspecto final del entorno de simulación desarrollado empleando la herramienta de *software* MatLab/Simulink

10 Figura 34: Trayectoria cartesiana de prueba

Figura 35: Trayectoria cartesiana: vista x-y-z deseada y real (controlador neuronal)

Figura 36: Trayectoria cartesiana: vista x-y deseada y real (controlador neuronal)

Figura 37: Trayectoria cartesiana: vista y-z deseada y real (controlador neuronal)

Figura 38: Trayectoria articular: deseada y real (controlador neuronal)

15 Figura 39: Fuerzas y torques aplicados (controlador neuronal)

Figura 40: Error de la trayectoria cartesiana (controlador neuronal)

Figura 41: Error de la trayectoria articular (controlador neuronal)

Figura 42: índice de desempeño correspondiente a la trayectoria articular

Figura 43: índice de desempeño correspondiente a la trayectoria cartesiana

Figura 44: Imagen de la implementación real de un robot redundante: vista en perspectiva, eslabones rotatorios inclinados

Figura 45: Imagen de la implementación real de un robot redundante: vista de perfil, eslabones rotatorios inclinados.

5

Descripción detallada de la invención

La metodología de diseño e implementación de un simulador gráfico para robots industriales redundantes, de acuerdo a la presente invención, contempla la utilización de herramientas de software, como por ejemplo: MatLab/Simulink, desarrollada por MathWorks, y MathType, desarrollada por Design Science, proporcionando los modelos cinemático y dinámico de robots industriales redundantes. Se describe el procedimiento necesario para elaborar, a través de funciones embebidas, un simulador gráfico que permite ensayar dicho tipo de robots, incluyendo: trayectorias de prueba tanto cartesianas como articulares, actuadores y controlador. Esta metodología permite obtener los modelos cinemático y dinámico de este tipo de robots de forma simbólica y en colores, y proporciona una ejecución de la simulación de manera rápida y precisa al emplear funciones embebidas de Simulink, por ejemplo. En la actualidad, no se encuentran simuladores dedicados al ensayo de robots industriales redundantes de N GDL con las características señaladas.

Descripción del Método

La presente invención describe el método para simular, obteniendo gráficamente las correspondientes curvas de desempeño, robots manipuladores industriales de N GDL.

5 La metodología propuesta contempla las siguientes etapas:

- Definir los sistemas de ejes coordenados para un robot de N GDL, aplicando el algoritmo de Denavit y Hartenberg Estándar.
- Generar las siguientes funciones:

- 10
 - o Función que entregue una base de datos simbólicos que se emplearán para iniciar el cálculo de la cinemática en un robot.
 - o Función que entregue una base de datos simbólicos que se emplearán para iniciar el cálculo de la dinámica en un robot.
 - o Función que permita calcular el modelo dinámico en un robot utilizando el método de balance de energía.
- 15
 - o Función que modifique símbolos, en una expresión cinemática o dinámica simbólica, adecuándolos para realizar una conversión a LaTeX (ref. 7).
 - o Función que reemplace constantes y variables, en una expresión simbólica, con símbolos en colores, y que proporcione el resultado en formato LaTeX.

- o Función que reemplace las funciones trigonométricas: sin y cos, y los símbolos theta, con símbolos "s" y "c" y "_", respectivamente.
 - o Función que simplifique la notación de expresiones matemáticas simbólicas.
- 5 - Ingresar el número de GDL N (sin restricciones), los tipos de articulaciones (rotacional o prismática), y programar las funciones que describen la cinemática y dinámica del robot.
- Ingresar, a las funciones, los parámetros correspondientes a las longitudes de los eslabones del robot, en unidades MKS (metros). A partir del ingreso de esta
- 10 información y de la cinemática conseguida, obtener el volumen de trabajo del robot.
- Escoger un software orientado a la programación gráfica y proceder a elaborar cuatro bloques fundamentales que componen el Simulador Gráfico:
- 1) Bloque Brazo: este componente se construye a través de cuatro subsistemas
- 15 para representar la matriz de inercia, el vector de fuerzas centrífugas y de Coriolis, el vector de fuerzas gravitacionales, y el vector de fuerzas de fricción.
- Estos cuatro subsistemas representan componentes del modelo dinámico del robot, como se detalla a continuación, tras aplicar el algoritmo de Denavit y Hartenberg Estándar:

$$\tau = \mathbf{M}(q)\ddot{q} + \mathbf{C}(q, \dot{q})\dot{q} + \mathbf{G}(q) + \mathbf{F}(\dot{q}) \quad (\text{Ec.1})$$

donde τ representa el vector de fuerzas generalizadas (de dimensión $N \times 1$), M es la matriz de inercia (de dimensión $N \times N$), C es el vector de fuerzas centrífugas y de Coriolis (de dimensión $N \times 1$), q representa los componentes del vector de posición de las articulaciones, \dot{q} representa los componentes del vector de velocidades de las articulaciones, G es el vector de fuerzas de gravitacionales (de dimensión $N \times 1$), F es el vector de fuerzas de fricción (de dimensión $N \times 1$) y \ddot{q} es el vector de aceleraciones de las articulaciones (de dimensión $N \times 1$).

- o Insertar, para cada uno de los cuatro subsistemas señalados, bloques con las funciones que describen la respectiva dinámica del robot, es decir: la matriz de inercia, el vector de fuerzas centrífugas y de Coriolis, el vector de fuerzas de gravitacionales, y el vector de fuerzas de fricción.
- o Ordenar los cuatro subsistemas y agruparlos en un subsistema mayor incorporando un bloque denominado "Constantes" en el cual se especifican los parámetros constantes que caracterizan el robot, como: masa y longitud de eslabones, valor de los momentos de inercia, etc.
- o Realizar un enmascaramiento del subsistema mayor e incluir una interfaz para ingresar las especificaciones de los parámetros constantes que caracterizan el robot. El enmascaramiento cumple la función de proteger datos sensibles, de manera que no puedan ser eliminados fácilmente por el usuario a causa de un error de tipeo, por ejemplo (ref. 8).

2) Bloque Actuadores: crear N subsistemas para representar los N actuadores del robot, adicionando N bloques de funciones que modelen el comportamiento dinámico de cada actuador del robot.

- o Realizar un enmascaramiento de los N subsistemas.

5 o Programar una interfaz de usuario que permita ingresar los parámetros de los N actuadores del robot.

3) Bloque Controlador: crear N subsistemas para representar los N controladores del robot, adicionando N bloques de funciones que modelen el comportamiento dinámico de cada controlador del robot.

10 o Ordenar y agrupar los N subsistemas en un bloque llamado "Controlador" en un único subsistema mayor.

- o Realizar un enmascaramiento del bloque llamado "Controlador".

- o Programar una interfaz de usuario que permita ingresar los parámetros de los N controladores del robot.

15 4) Bloque Trayectoria: crear subsistemas, sin restricción de cantidad, dependiendo de las diversas trayectorias generadas por los diversos GDL del robot que el usuario desee visualizar. Esto se logra al incorporar las correspondientes trayectorias articulares que se utilizan para ensayar el robot.

20 o Emplear subsistemas para insertar los bloques necesarios y establecer los recorridos deseados.

- o Asociar cada bloque de trayectoria a tres archivos de datos que contengan la información de: posición; velocidad y aceleración de las articulaciones.
 - o Realizar un enmascaramiento del bloque llamado "Trayectoria".
 - o Programar una interfaz de usuario que permita seleccionar la trayectoria
- 5 que se desea probar. Comprobar el funcionamiento y respuesta de cada uno de los bloques y realizar acciones correctivas hasta obtener el resultado deseado.

Cada bloque funcionará bien si el *software* en el cual se programó, por ejemplo MatLab/Simulink, no despliega ninguna advertencia *{warning}* o mensaje de error

10 cuando se selecciona la acción de simulación del bloque aludido. Estos errores, en general, son de convergencia, es decir que las simulaciones entreguen valores reales y acotados (que no tiendan a infinito). La respuesta deseada de cada bloque se refiere a que éste entregue una salida de dimensiones esperadas (descritas anteriormente) ya que si no hay errores en las simulaciones la respuesta final,

15 entregada por el bloque, será correcta. Acciones correctivas se refiere a verificar el correcto ingreso de los datos para las simulaciones, así el resultado de los bloques serán los deseados.

Finalmente, ensamblar cada uno de los bloques señalados (Brazo, Actuadores, Controlador y Trayectoria) generando de esta manera un simulador que permita

20 ensayar un robot de *N* GDL.

La metodología propuesta, se resume a través del diagrama de flujo que se muestra en la Figura 1.

Las simulaciones de los bloques individuales será correcta si tras ingresados los parámetros que caracterizan el robot a simular, y realizar la simulación de éste, los resultados de cada subsistema por separado, *i.e.*, matriz de inercia, vector de fuerzas de fricción, vector de fuerzas centrífugas y de Coriolis y vector de fuerzas gravitacionales, entregan resultados que convergen a funciones numéricas con dimensiones espaciales consistentes (matriz de dimensión $N \times 1$ y vectores de dimensiones $N \times 1$), según se explicó anteriormente a través de la descripción de la ecuación 1.

Ejemplo de Aplicación del método de simulación

A continuación se describen los pasos para simular, obteniendo gráficamente las correspondientes curvas de desempeño, un robot industrial redundante de 5 GDL, como el que se muestra en la Figura 2.

En primer lugar se obtienen los modelos cinemático y dinámico. De esta forma, se plantean los sistemas de ejes coordenados, aplicando el algoritmo de Denavit y Hartenberg Estándar, como se parecía en la Figura 3.

A continuación, empleando la herramienta de software MatLab, se desarrollan las siguientes funciones:

- datos_dinJk(n): función que genera una base de datos simbólicos que se utilizan para calcular la dinámica del robot (Figura 4).

- `dinamicajk()`: función que permite calcular el modelo dinámico del robot utilizando el método de balance de energía (Figura 5).
- `modJk()`: función que modifica símbolos, en una expresión cinemática o dinámica simbólica, adecuándolos para a una conversión LaTeX (Figura 6).
- 5 - `latexjk(in)`: función que reemplaza constantes y variables, en una expresión simbólica "in", con símbolos en colores, y proporciona el resultado en formato LaTeX (Figura 7).
- `simpjk(in)`: función que reemplaza las funciones: sin y eos, y los símbolos theta, con símbolos "s" y "c" y "_", respectivamente (Figura 8).
- 10 - `convjk(in)`: función que convierte y simplifica la notación de expresiones matemáticas simbólicas, en una expresión simbólica de entrada "in" (Figura 9).

Con las funciones Matlab ya creadas, se programan los Script MatLab necesarios para obtener los modelos cinemático y dinámico del robot redundante, como se indica en la Figura 10 y en la Figura 11, respectivamente. Deben considerarse: el

15 número de GDL (sin restricciones) y el tipo de articulación (rotacional o prismática).

Posteriormente, a partir de la cinemática obtenida y empleando su respectivo Script, se obtiene el espacio de trabajo del robot redundante. En este punto, se ingresan al programa los datos correspondientes a las longitudes de los eslabones del robot en unidades de metros.

20 En la Figura 12 se aprecia un ejemplo del resultado del espacio de trabajo obtenido en el plano xy para el robot redundante de 5 GDL, considerando la longitud de los

eslabones rotatorios de 0,2 m. En la Figura 13 se muestra una vista isométrica del espacio de trabajo bosquejado a partir de la información que se obtiene del espacio de trabajo desarrollado con el Script MatLab para calcular la cinemática del robot redundante.

5 Se procede a continuación, mediante la herramienta de programación Simulink, a elaborar cuatro bloques fundamentales que componen el Simulador Gráfico.

1) Bloque Brazo: este componente se construye a través de subsistemas Simulink para representar la matriz de inercia, el vector de fuerzas de fricción, el vector de fuerzas centrífugas y de Coriolis, y el vector de fuerzas gravitacionales, como se
10 indica en la Figura 14.

Se agrupan los cuatro subsistemas señalados en un subsistema mayor y se escriben explícitamente las funciones que describen la dinámica del robot, como se observa en la Figura 15 y en la Figura 16, respectivamente.

Luego de programar las funciones embebidas, los subsistemas Simulink
15 señalados se ordenan y se agrupan en un subsistema Simulink mayor incorporando un bloque denominado "Constantes" en el cual se especifican los datos del robot redundante, como se indica en la Figura 17.

En esta etapa, se procede a realizar un enmascaramiento del subsistema Simulink mayor (Figura 18) en donde se incluye una interfaz para ingresar las
20 especificaciones de las constantes del robot redundante, como se señala en la Figura 19.

2) Bloque Actuadores: utilizando subsistemas Simulink se representan los actuadores del manipulador robotizado adicionando el correspondiente bloque de función embebida y programando su algoritmo asociado, como se presenta en la Figura 20 y en la Figura 21, respectivamente.

5 Posteriormente, se elabora un enmascaramiento del subsistema Simulink (Figura 22) en donde se programa una interfaz de usuario que permite ingresar los datos de los actuadores del robot redundante, como se muestra en la Figura 23.

3) Bloque Controlador: en este bloque se emplean subsistemas Simulink que componen la estructura del controlador que se utilizará para realizar la
10 simulación, como se indica en la Figura 23. En este caso, de manera particular, se muestra el desarrollo de la estructura de un controlador par calculado con funciones neuronales ("Par Cale - Neuronal")

A continuación se insertan los correspondientes bloques de funciones embebidas para cada uno de los subsistemas pertenecientes a la estructura del controlador y
15 se programan sus correspondientes algoritmos, como se observa en la Figura 25 y en la Figura 26, respectivamente.

En este punto, los subsistemas Simulink componentes del bloque "Controlador" se ordenan y se agrupan en un único subsistema Simulink agregando un bloque denominado "Saturación" que permite acotar las señales de corrección que
20 proporciona el controlador del robot redundante, como se indica en la Figura 27.

Se continúa con un enmascaramiento del subsistema controlador, como se señala en Figura 28, en donde se considera una interfaz para ingresar las especificaciones de las ganancias.

4) Bloque Trayectoria: este componente se desarrolla para incorporar las correspondientes trayectorias articulares que se utilizan para ensayar el robot redundante. Al igual que en los casos anteriores, se emplean subsistemas Simulink para insertar los bloques necesarios y establecer los recorridos deseados, sin restricción en su cantidad. A modo de ejemplo, en la Figura 29, se consideran tres trayectorias.

10 Cada bloque de trayectoria se asocia a tres archivos de datos que contienen la información de: posición; velocidad y aceleración de las articulaciones, como se aprecia en la Figura 30.

15 Luego, se realiza un enmascaramiento del subsistema trayectoria, como se muestra en la Figura 31, en donde se contempla además una interfaz para seleccionar la trayectoria que se desea probar (Figura 32).

20 Las simulaciones de los bloques individuales será correcta si tras ingresados los parámetros que caracterizan el robot a simular, *i.e.*, longitudes y masas de sus eslabones, momentos de inercia, etc., y realizar la simulación de éste, los resultados de cada subsistema por separado, *i.e.*, matriz de inercia, vector de fuerzas de fricción, vector de fuerzas centrífugas y de Coriolis y vector de fuerzas

gravitacionales, entregan resultados que convergen a funciones numéricas con dimensiones espaciales consistentes (matriz de dimensión 5×1 y vectores de dimensiones 5×1 , según corresponda).

Finalmente, se procede a ensamblar cada uno de los bloques señalados (Brazo, 5 Actuadores, Controlador y Trayectoria) generando de esta manera un simulador que permite ensayar un robot redundante de 5 GDL. El aspecto final del simulador se observa en la Figura 33.

Los resultados obtenidos mediante esta metodología, se muestran a continuación. Se utiliza una trayectoria cartesiana de prueba graficada con los Script MatLab, que 10 consiste en una espiral en el espacio cartesiano, como se exhibe en la Figura 34. Se emplea para este ensayo un controlador neuronal.

Las comparaciones entre la trayectoria cartesiana deseada y la trayectoria cartesiana real, de acuerdo al comportamiento del controlador neuronal, se indican en las Figura 35, 36 y 37, respectivamente. En ellas, se muestran las gráficas 15 pertinentes en: el espacio, el plano x-y y el plano y-z, correspondientemente.

En la Figura 8 y 39, se indican las gráficas relacionadas a las trayectorias articulares deseadas y reales, y a las fuerzas y torques aplicados, mediante el uso del controlador neuronal, correspondientemente.

Las gráficas de los errores producidos entre las trayectorias: cartesiana deseada y 20 real, y articular deseada y real, aplicando el controlador neuronal, se señalan a través de la Figura 40 y 41, respectivamente.

La metodología de la presente invención permite desarrollar un simulador para un robot redundante de 5 GDL, en el cual pueden ensayarse diferentes controladores y/o diferentes trayectorias de prueba. Además, como una característica destacada, es posible ensayar otros manipuladores con diferentes estructuras y con mayores
5 grados de libertad, quedando limitada esta última característica solo a las capacidades de procesamiento del sistema computacional en donde se ejecute el simulador.

Ventajas y desventajas con respecto a la técnica anterior

Es importante señalar que la metodología propuesta admite el desarrollo de
10 simuladores para otros tipos de manipuladores robotizados, gracias a que proporciona los modelos cinemático y dinámico, para robots de diferentes características estructurales, que pueden ingresarse a Simulink a través de funciones embebidas que permiten realizar una (o algunas) tareas específicas dependiendo del tipo de manipulador robotizado a simular.

15 La metodología propuesta, sin embargo, requiere de manera imprescindible para su implementación de los software MatLab/Simulink y MathType.

A modo de ejemplo, a continuación se muestra la expresión resultado del modelo cinemático obtenido para un robot redundante de 7 GDL (configuración del efector terminal roll-pitch),

$$\mathbf{T} = \begin{bmatrix} T_{11} & T_{12} & T_{13} & T_{14} \\ T_{21} & T_{22} & T_{23} & T_{24} \\ T_{31} & T_{32} & T_{33} & T_{34} \\ T_{41} & T_{42} & T_{43} & T_{44} \end{bmatrix}$$

$$T_{11} = -\{ \sin(\theta_4 + \theta_4 + \theta_4) \} \{ \sin(\theta_6) - \cos(\theta_6) \} + \cos(\theta_4 + \theta_4 + \theta_4) \sin(\theta_4) \} \sin(\theta_6)$$

$$T_{12} = \{ \sin(\theta_2 + \theta_3 + \theta_3) \{ \sin(\theta_6) - \cos(\theta_6) \} + \cos(\theta_2 + \theta_3 + \theta_3) \sin(\theta_6) \} \cos(\theta_6)$$

$$T_{13} = -\sin(\theta_2 + \theta_3 + \theta_3) \{ \sin(\theta_6) + \cos(\theta_6) \} - \cos(\theta_2 + \theta_3 + \theta_3) \cos(\theta_6)$$

$$\theta_4 = l_2 \cos(\theta_6) + l_3 \cos(\theta_2 + \theta_3) + l_4 \cos(\theta_2 + \theta_3 + \theta_3) - l_7 \sin(\theta_7) \{ \sin(\theta_2 + \theta_3 + \theta_3) \{ \sin(\theta_6) - \cos(\theta_6) \} + \cos(\theta_2 + \theta_3 + \theta_3) \sin(\theta_6) \}$$

$$L_{21} = \{ \cos(\theta_6 + \theta_4 + \theta_4) \{ \sin(\theta_6) - \cos(\theta_6) \} - \sin(\theta_2 + \theta_3 + \theta_3) \sin(\theta_4) \} \sin(\theta_6)$$

$$T_{22} = -\{ \cos(\theta_2 + \theta_3 + \theta_3) \{ \sin(\theta_6) - \cos(\theta_6) \} - \sin(\theta_2 + \theta_3 + \theta_3) \sin(\theta_6) \} \cos(\theta_6)$$

$$\theta_4 = \cos(\theta_4 + \theta_4 + \theta_4) \{ \sin(\theta_6) + \cos(\theta_6) \} - \sin(\theta_2 + \theta_3 + \theta_3) \cos(\theta_6)$$

$$T_{34} = l_1 \sin(\theta_7) + l_2 \sin(\theta_2 + \theta_3) + l_4 \sin(\theta_2 + \theta_3 + \theta_3) + l_7 \sin(\theta_7) \{ \cos(\theta_2 + \theta_3 + \theta_3) \{ \sin(\theta_6) - \cos(\theta_6) \} - \sin(\theta_2 + \theta_3 + \theta_3) \sin(\theta_6) \}$$

$$T_{31} = -\cos(\theta_7)$$

$$T_{32} = -\sin(\theta_7)$$

$$T_{33} = 0$$

$$T_{34} = l_1 + \theta_1 - \theta_1 - l_2 - l_3 - l_7 \cos(\theta_7)$$

$$T_{41} = T_{42} = T_{43} = 0$$

$$T_{44} = 1$$

y a continuación se presenta una expresión correspondiente a la matriz de inercia

como parte del modelo dinámico de un robot redundante de 7 GDL (configuración

5 del efector terminal pitch-roll).

$$\mathbf{M} = \begin{bmatrix} \mathbf{X} & M_{12} & v_{13} & M_{14} & M_{15} \\ M_{12} & M_{22} & M_{23} & M_{24} & M_{25} \\ M_{21} & 3/4 & M_{32} & M_{34} & M_{35} \\ M_{41} & M_{42} & M_{43} & M_{44} & M_{45} \\ M_{51} & M_{52} & M_{53} & M_{54} & M_{55} \end{bmatrix}$$

$$M_{11} = m_1 + m_2 + m_3 + m_4 + m_5$$

$$M_{12} = M_{13} = M_{14} = 0$$

$$M_{15} = -m_3$$

$$M_{21} = M_{12}$$

$$M_{22} = l_{c2}^2 m_2 + (l_2^2 + l_{c2}^2 + 2l_2 l_{c2} \cos \theta_2) m_4 + I_{2zz} + J_{2zz} + I_{4zz} + \dots$$

$$(l_2^2 + l_3^2 + l_{c4}^2 + 2l_2 l_3 \cos \theta_3 + 2l_2 l_{c4} \cos \theta_3 + 2l_3 l_{c4} \cos(\theta_3 + \theta_4)) m_4 + \dots$$

$$(l_2^2 + l_3^2 + l_4^2 + 2l_2 l_3 \cos \theta_3 + 2l_2 l_4 \cos \theta_3 + 2l_3 l_4 \cos(\theta_3 + \theta_4)) m_5$$

$$M_{23} = (l_{c3}^2 + l_3 l_{c3} \cos \theta_3) m_3 + I_{3zz} + I_{4zz} + \dots$$

$$(l_3^2 + l_{c4}^2 + l_3 l_4 \cos \theta_4 + 2l_3 l_{c4} \cos \theta_4 + 3/4 l_{c4}^2 (3/4 + 3/4)) m_4 + \dots$$

$$(3/4^2 + l_{c4}^2 + l_3 l_4 \cos \theta_4 + 2l_3 l_{c4} \cos \theta_4 + l_3 l_4 \cos(\theta_3 + 3/4)) m_5$$

$$M_{24} = (l_{c4}^2 + l_3 l_{c4} \cos \theta_4 + l_3 l_{c4} \cos(\theta_3 + \theta_4)) m_4 + \dots$$

$$(l_4^2 + l_3 l_4 \cos \theta_4 + l_3 l_4 \cos(\theta_3 + \theta_4)) m_5 + I_{4zz}$$

$$M_{25} = 0$$

$$3/4 = 3/4$$

$$M_{32} = M_{23}$$

$$M_{33} = l_{c3}^2 m_3 + (l_3^2 + l_{c4}^2 + 2l_3 l_{c4} \cos \theta_4) m_4 + \dots$$

$$(l_3^2 + l_4^2 + 2l_3 l_4 \cos \theta_4) m_5 + I_{3zz} + I_{4zz}$$

$$M_{34} = (l_{c4}^2 + l_3 l_{c4} \cos \theta_4) m_4 + (l_4^2 + l_3 l_4 \cos \theta_4) m_5 + I_{4zz}$$

$$M_{35} = 0$$

$$M_{41} = M_{14}$$

$$M_{42} = M_{24}$$

$$M_{43} = M_{34}$$

$$M_{44} = l_{c4}^2 m_4 + l_4^2 m_5 + I_{4zz}$$

$$M_{45} = 0$$

$$M_{51} = M_{15}$$

$$3/4 = M_{25}$$

$$M_{55} = M_{55}$$

$$M_{54} = M_{45}$$

$$M_{55} = m_5$$

A partir de la metodología presentada, se han elaborado diferentes simuladores en función de distintos algoritmos de control con el objetivo de ensayar un robot industrial redundante de 5 GDL tipo SCARA. Como una muestra de los logros que pueden obtenerse, en la Figura 42 y 43, se presenta un resumen de los resultados de los ensayos del robot redundante empleando siete controladores (sinh-consh: seno coseno hiperbólico, md: modo deslizante, md-tanh: modo deslizante hiperbólico, par-cal: par calculado, aprend: adaptivo con aprendizaje, adaptivo: inercia adaptativa y neuronal) señalando índices de desempeño de los errores articulares y cartesianos, a través de valores RMS (Root Mean Square), respectivamente.

La metodología expuesta, ha permitido en particular la implementación real de un nuevo robot industrial redundante de 5 GDL. Para ello, se han obtenido los modelos cinemático dinámico del robot y a partir de éstos se ha elaborado el simulador correspondiente. Dicho simulador se ha ejecutado en un computador Intel Core i5-4430 8 GB RAM con el software MatLab en sus versiones r2009a y r2013a permitiendo ensayar, sin bloqueos ni detenciones del proceso, el robot para diferentes trayectoria articulares y longitudes de los eslabones. Los resultados conseguidos, luego de los análisis respectivos, han hecho posible la implementación de un robot redundante de 5 GDL tipo SCARA, como se aprecia en la Figura 44 y 45.

REFERENCIAS.

- (1) HEMERO: Herramienta MatLab/Simulink para el estudio de manipuladores y robots móviles (2006), J. Iván Maza Alcañiz y Aníbal Ollero Baturone,
5 <http://grvc.us.es/hemero/descargar.html>.
- (2) SpaceLib: (2009), Giovanni Legnani, MatLab Central File Exchange,
<http://www.mathworks.com/matlabcentral/fileexchange/1657-spacelib>
- (3) Robotics Toolbox: P.I. Corke, "Robotics, Vision & Control", Springer (2011), ISBN 978-3-642-20143-1, (http://petercorke.com/Toolbox_software.html)
- 10 (4) NXT SCARA: Two-Link Planar Robot Arm, Controller Design (2009), Yorihiisa Yamamoto, MatLab Central File Exchange,
<http://www.mathworks.com/matlabcentral/fileexchange/22126-nxt-scara~two-link-planar-robot-arm~controller-design>
- (5) Model Based Robotic using MATLAB, Simulink, and the FANUC arm, First
15 submitted by MathWorks Classroom Resources Team (2013), Matlab Central File Exchange,
<https://www.mathworks.com/matlabcentral/linkexchange/links/3688-model-based-robotic-control-design-using-matlab-simulink-and-the-fanuc-arm>
- (6) SCARA with Path trajectory: (2014), Mohammed Elamassie, Matlab Central File
20 Exchange, <http://www.mathworks.com/matlabcentral/fileexchange/45365-scara-wjth-j3ath:trajectory>.
- (7) <http://es.wikipedia.org/wiki/LaTeX>.
- (8) http://es.wikipedia.org/wiki/Enriascara:niento_de_datos

REIVINDICACIONES

1. Método para simular de manera gráfica en colores movimientos de robots Industriales redundantes, **CARACTERIZADO** porque comprende las etapas de:

definir los sistemas de ejes coordenados para N Grados De Libertad (GDL);

5 aplicar algoritmos de Denavit y Hartenberg estándar a los ejes de coordenadas;

generar funciones unitarias para calcular la cinemática y dinámica del robot;

ejecutar las funciones utilitarias, mediante algoritmos, para obtener los modelos cinemáticos y dinámicos del robot;

10 ingresar los datos específicos del robot, de acuerdo al modelo cinemático;

resolver la ecuación cinemática y dinámica del robot y obtener las representación gráfica de los movimientos del robot;

generar un conjunto de bloques para la simulación gráfica, en donde cada módulo consiste en:

15 - bloque brazo que comprende los siguientes subsistemas:

matriz de inercia;

vector de fuerzas de fricción;

vector de fuerzas centrífugas y de Coriolis; y

vector de fuerzas gravitacionales;

- módulo actuadores, para la representación de los actuadores que manipulan el robot;
- módulo controlador, para generar submódulos correspondientes a la estructura del controlador que utilizará el simulador; y
- 5 - módulo trayectoria, que incorpora las trayectorias articulares para el ensayo del robot;

ingresar señales de entrada a cada uno de los módulos;

ejecutar la simulación individual de cada uno de los módulos, en donde si el resultado está dentro de lo esperado, se aprueba el módulo, de lo contrario se
10 realizan acciones correctivas;

ensamblar los módulos, una vez que han pasado la etapa de simulación individual;

seleccionar una trayectoria para simular el movimiento del robot; y

detectar anomalías en la simulación.

15 2. El método de acuerdo a la reivindicación 1, **CARACTERIZADO** porque la simulación gráfica se realiza mediante la utilización de herramientas de software MatLab/Simulink y MathType.

3. El método de acuerdo a la reivindicación 1, **CARACTERIZADO** porque las funciones utilitarias se ejecutan mediante la herramienta de software Simulink.

4. El método de acuerdo a la reivindicación 1, CARACTERIZADO porque ejecutar la simulación individual de cada uno de los bloques se realiza mediante la utilización de la herramienta de software MatLab.

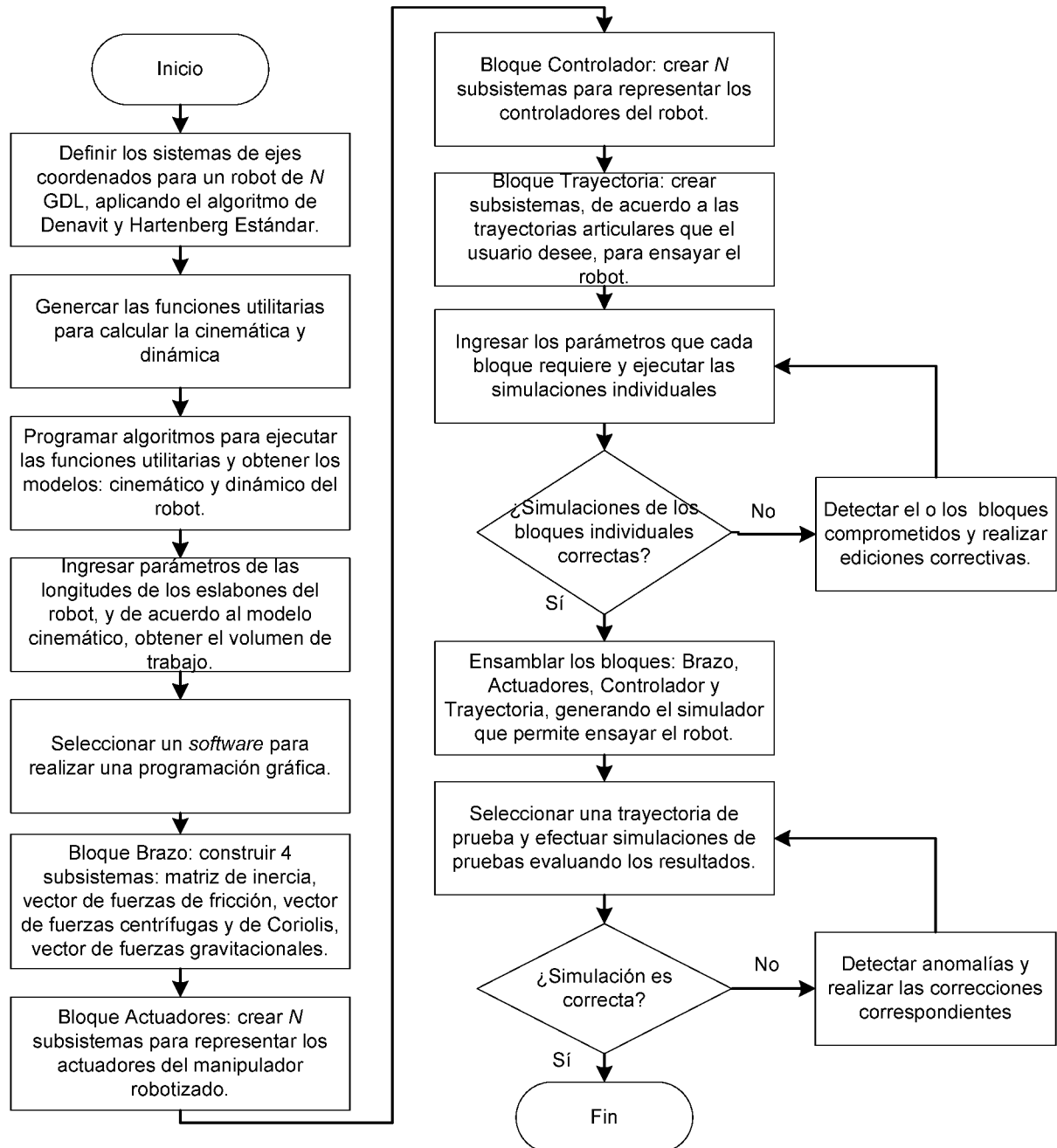


FIG. 1

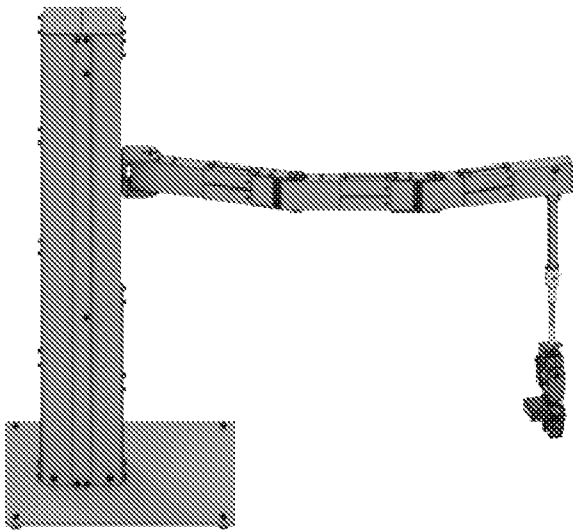


FIG. 2

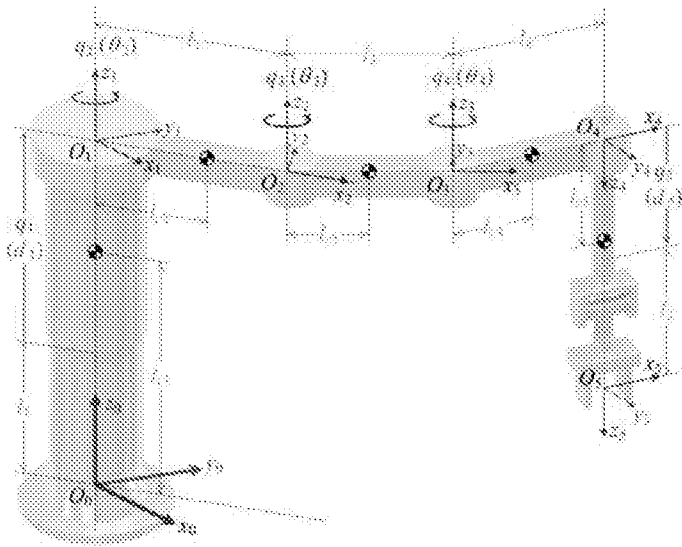


FIG. 3

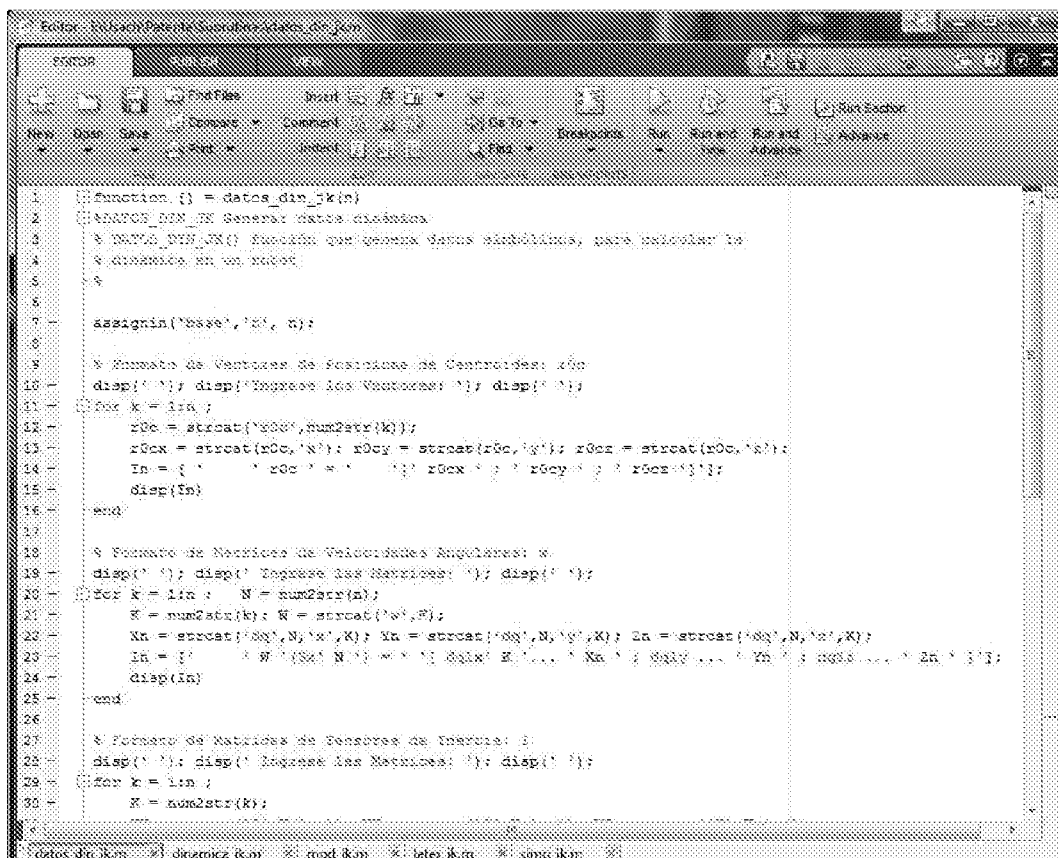
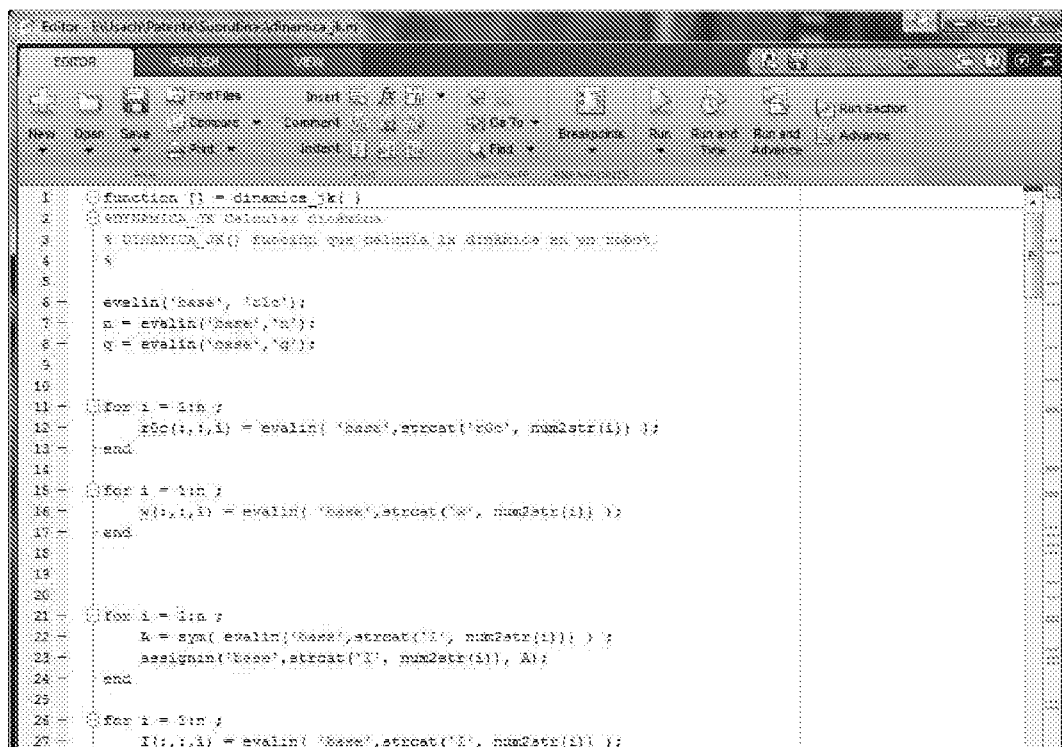


FIG. 4



```

1 function [] = dinamica_1k()
2 % DINAMICA_1K Calcula la dinamica
3 % DINAMICA_1K() funciona que calcula la dinamica en un robot.
4 %
5
6 - evalin('base','x0');
7 - n = evalin('base','n');
8 - g = evalin('base','g');
9
10
11 - for i = 1:n;
12 -     x0(i,1,i) = evalin('base','x0cat','num2str(i)');
13 - end
14
15 - for i = 1:n;
16 -     x(i,1,i) = evalin('base','xcat','num2str(i)');
17 - end
18
19
20
21 - for i = 1:n;
22 -     k = sym(evalin('base','xcat','num2str(i)'));
23 -     assignin('base','xcat','num2str(i)', A);
24 - end
25
26 - for i = 1:n;
27 -     I(i,1,i) = evalin('base','xcat','num2str(i)');

```

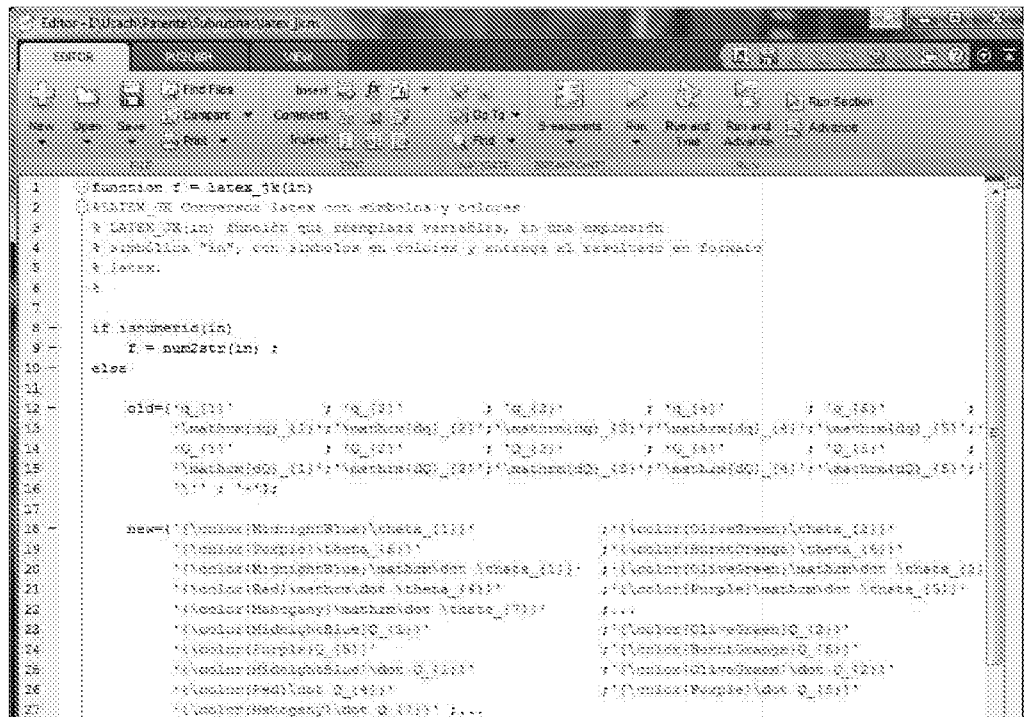
FIG. 5


```

1 function [] = mod_k()
2 %MOD_K Modifica a matriz
3 % MOD_K() function que modifica a matriz, em uma expressão simbólica,
4 % para adequar a uma conversão para
5 %
6
7 % sym('1_1 1_2 1_3 1_4 1_5 1_6 1_7 real');
8 % sym('1_2 1_3 1_4 1_5 1_6 1_7 real');
9 % sym('1_3 1_4 1_5 1_6 1_7 real');
10 % sym('1_4 1_5 1_6 1_7 real');
11 % sym('1_5 1_6 1_7 real');
12 % sym('1_6 1_7 real');
13 % sym('1_7 real');
14 % sym('1 0 0 0 0 0 0 real');
15
16
17 for i=1:7
18     A = sym('A', num2str(i), 'real') ; assignin('base', A, sym(A, 'real'));
19     B = sym('B', num2str(i), 'real') ; assignin('base', char(B), B);
20     assignin('base', char(A), B);
21
22     A = sym('A', num2str(i), 'real') ; assignin('base', A, sym(A, 'real'));
23     B = sym('B', num2str(i), 'real') ; assignin('base', char(B), B);
24     assignin('base', char(A), B);
25
26     A = sym('A', num2str(i), 'real') ; assignin('base', A, sym(A, 'real'));
27     B = sym('B', num2str(i), 'real') ; assignin('base', char(B), B);

```

FIG. 6

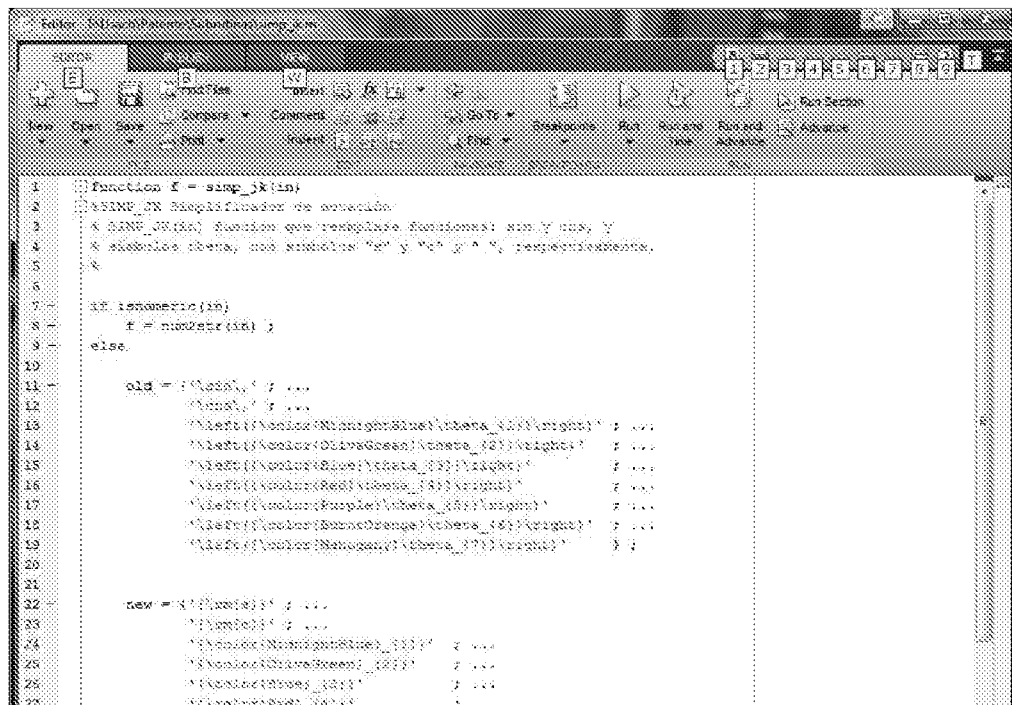


```

1 function f = latex_jk(in)
2 %LATEX_JK Conversor latex con simbolos y colores
3 % LATEX_JK(in) función que reemplaza variables, en una expresión
4 % simbólica "in", con símbolos en colores y entrega el resultado en formato
5 % latex.
6 %
7
8 if isnumeric(in)
9     f = num2str(in);
10 else
11
12     old = {'Q_11'}; % 'Q_12'}; % 'Q_13'}; % 'Q_14'}; % 'Q_15'}; %
13 %\mathrel{Q_11} %\mathrel{Q_12} %\mathrel{Q_13} %\mathrel{Q_14} %\mathrel{Q_15} %
14 %Q_11 %Q_12 %Q_13 %Q_14 %Q_15 %
15 %\mathrel{Q_11} %\mathrel{Q_12} %\mathrel{Q_13} %\mathrel{Q_14} %\mathrel{Q_15} %
16 %' %';
17
18 new = {'\color{NinightBlue}\theta_1'}; %\color{OliveGreen}\theta_2'};
19 %\color{Purple}\theta_3'}; %\color{BurntOrange}\theta_4'};
20 %\color{NinightBlue}\mathbin{\dot{\theta}}_1'}; %\color{OliveGreen}\mathbin{\dot{\theta}}_2'};
21 %\color{Red}\mathrel{\dot{\theta}}_3'}; %\color{Purple}\mathbin{\dot{\theta}}_4'};
22 %\color{NinightBlue}\mathbin{\dot{\theta}}_5'}; %... %\color{OliveGreen}Q_12} %
23 %\color{Purple}Q_13} %\color{BurntOrange}Q_14} %
24 %\color{OliveGreen}\dot{Q}_11} %\color{OliveGreen}\dot{Q}_12} %
25 %\color{Red}\dot{Q}_13} %\color{Purple}\dot{Q}_14} %
26 %\color{NinightBlue}\dot{Q}_15} %... %\color{Purple}\dot{Q}_15} %
27

```

FIG. 7



```

1 function f = simp_jk(in)
2 %SIMP_JK Simplificador de expresión
3 % SIMP_JK(in) función que reemplaza funciones: sin y cos, y
4 % símbolos theta, con símbolos "s" y "c" y "s" y "c", respectivamente.
5 %
6
7 if isnumeric(in)
8     f = num2str(in);
9 else
10
11     old = {'sin'}; % ...
12     %cos % ...
13     %\left(\color{NinightBlue}\theta_1\right) % ...
14     %\left(\color{OliveGreen}\theta_2\right) % ...
15     %\left(\color{Purple}\theta_3\right) % ...
16     %\left(\color{Red}\theta_4\right) % ...
17     %\left(\color{Purple}\theta_5\right) % ...
18     %\left(\color{BurntOrange}\theta_6\right) % ...
19     %\left(\color{NinightBlue}\theta_7\right) % ...
20
21
22 new = {'\sin'}; % ...
23 %\cos % ...
24 %\color{NinightBlue}\theta_1 % ...
25 %\color{OliveGreen}\theta_2 % ...
26 %\color{Purple}\theta_3 % ...
27 %\color{Red}\theta_4 % ...

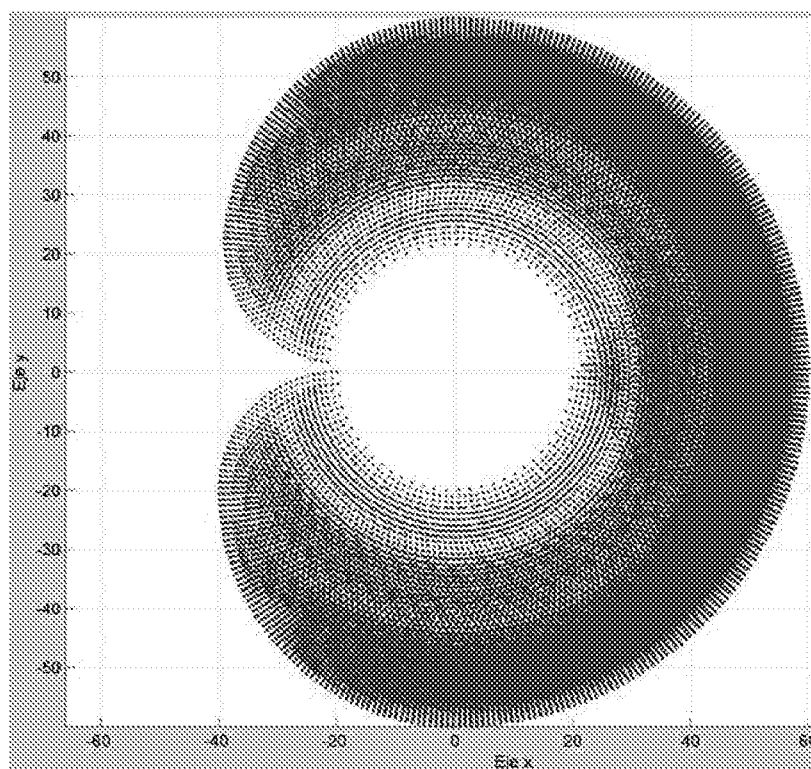
```

FIG. 8


```

1 % SIMULACION MANIPULADOR REDUNDANTE TIPO SCARA DE 5 GDL
2 % Cálculo de la dinámica de un manipulador redundante de tipo SCARA de 5
3 % grados de libertad: PEIRP.
4 %
5 % NOMBRE DEL MANIPULADOR
6 % Plantamiento de los estados de este controlador.
7 %
8 % <Dir_Ext_5_GDL_>.
9 %
10 % Se utiliza el método de "Newell-Hartenberg Denavit"
11
12 clear all; clc;
13
14 datos_dir_5(5)
15
16 r0c1 = 10 ; 0
17 r0c2 = (10*cos(q2) ; 10*sin(q2)
18 r0c3 = (12*cos(q2)+10*cos(q2+q3) ; 12*sin(q2)+10*sin(q2+q3)
19 r0c4 = (12*cos(q2)+13*cos(q2+q3)+10*cos(q2+q3+q4) ; 12*sin(q2)+13*sin(q2+q3)+10*sin(q2+q3+q4)
20 r0c5 = (12*cos(q2)+13*cos(q2+q3)+14*cos(q2+q3+q4) ; 12*sin(q2)+13*sin(q2+q3)+14*sin(q2+q3+q4)
21
22 w1 = [0 0 0 0 ; 0 0 0 0 ; 0 0 0 0]
23 w2 = [0 0 0 0 ; 0 0 0 0 ; 0 1 0 0]
24 w3 = [0 0 0 0 ; 0 0 0 0 ; 0 1 0 0]

```

**FIG. 12**

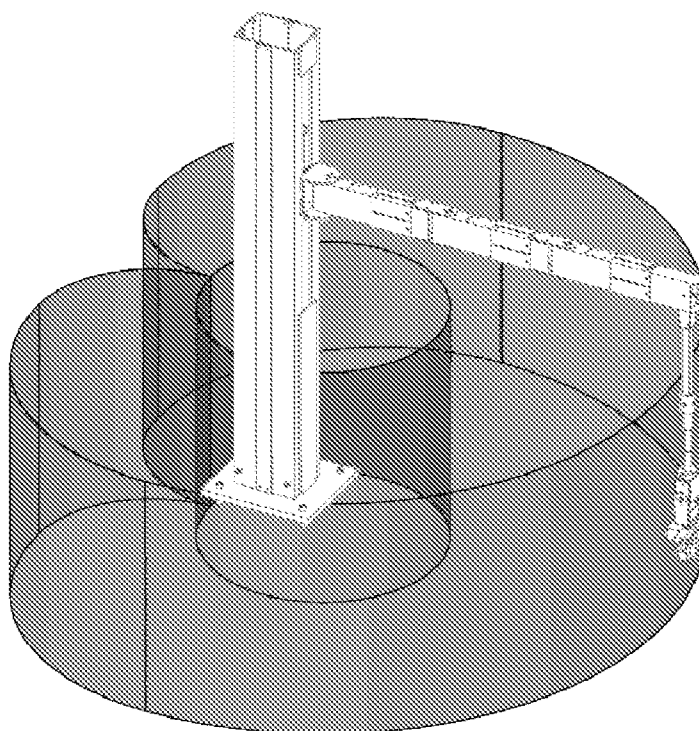


FIG. 13

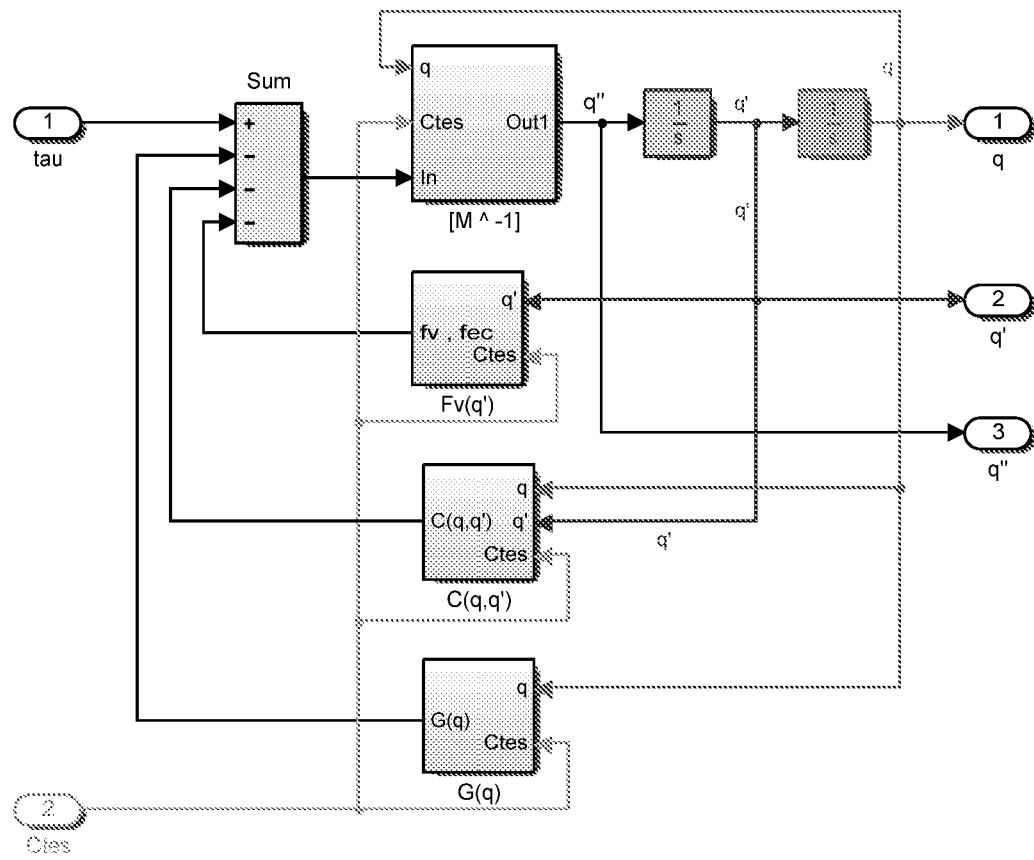


FIG. 14

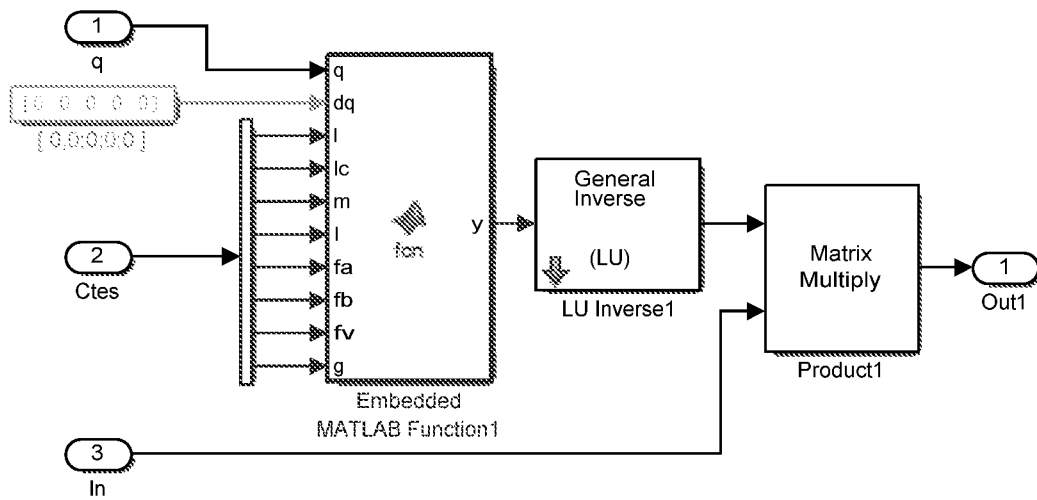


FIG. 15

```

Block editor: prop/palermo/Block Editor - MATLAB (2015) - Embedded MATLAB Function
2015/04/24 10:00:00
View
File Edit Insert View Tools Windows Help
New Open Save Copy Paste Comment Undo Redo Find Find Next Breakpoints Run Model Simulink Build Model Help
Block Editor
File Edit Insert View Tools Windows Help
1 function y = fon(q,dq,l,lc,m,l,fa,fb,fv,g)
2 % This block supports the Embedded MATLAB subset.
3 % See the help page for details.
4
5 q1 = q(1) ; q2 = q(2) ; q3 = q(3) ; q4 = q(4) ; q5 = q(5) ;
6 dq1 = dq(1) ; dq2 = dq(2) ; dq3 = dq(3) ; dq4 = dq(4) ; dq5 = dq(5) ;
7 l1 = l(1) ; l2 = l(2) ; l3 = l(3) ; l4 = l(4) ; l5 = l(5) ;
8 lc1 = lc(1) ; lc2 = lc(2) ; lc3 = lc(3) ; lc4 = lc(4) ; lc5 = lc(5) ;
9 m1 = m(1) ; m2 = m(2) ; m3 = m(3) ; m4 = m(4) ; m5 = m(5) ;
10 l1a = l1(1) ; l1a2 = l1(2) ; l1a3 = l1(3) ; l1a4 = l1(4) ; l1a5 = l1(5) ;
11 feo1a = fa(1) ; feo2a = fa(2) ; feo3a = fa(3) ; feo4a = fa(4) ; feo5a = fa(5) ;
12 feo1b = fb(1) ; feo2b = fb(2) ; feo3b = fb(3) ; feo4b = fb(4) ; feo5b = fb(5) ;
13 fv1 = fv(1) ; fv2 = fv(2) ; fv3 = fv(3) ; fv4 = fv(4) ; fv5 = fv(5) ;
14 gX = g(1) ; gY = g(2) ; gZ = g(3) ;
15
16 M11 = m1 + m2 + m3 + m4 + m5 ;
17 M12 = 0 ;
18 M13 = 0 ;
19 M14 = 0 ;
20 M15 = -m5 ;
21
22 M21 = M12 ;
23 M22 = (12^2 + 1*cos(q5)*12^13 + 2*cos(q3 + q4)*12^14 + 13^2 + 1*cos(q6)*13^13 + 14^2)*m5 + 12a2 + 13a2 ;
24 M23 = (13^2 + 1*cos(q6)*13^13 + 12*cos(q3)*13 + 14^2 + 12*cos(q3 + q4)*13)*m5 + 13a2 + 14a2 + (12^2 + 13^2) ;
25 M24 = (14^2 + 13*14*cos(q6) + 12*14*cos(q3 + q4))*m5 + 14a2 + (12^2 + 13*14*cos(q3) + 12*14*cos(q3 +
26 M25 = 0 ;
27

```

FIG. 16

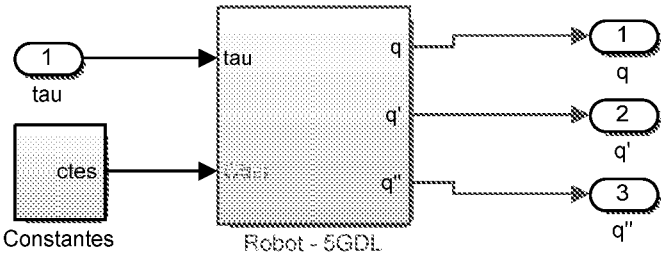


FIG. 17

5

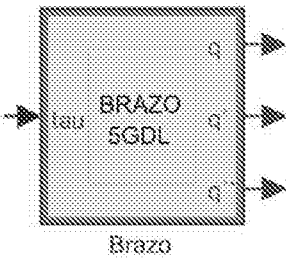


FIG. 18

10

Function Block Parameters: Brass

BRAZO (mask)

Parámetros Dinámicos

Parameters

[i1 ; i2 ; i3 ; i4 ; i5]

[0.54 ; 0.2 ; 0.2 ; 0.2 ; 0.14]

[k1 ; k2 ; k3 ; k4 ; k5]

[0.0229 ; 0.0229 ; 0.0229 ; 0.0229 ; 0.0983]

[m1 ; m2 ; m3 ; m4 ; m5]

[2.0458/2 ; 2.0458/2 ; 2.0458/2 ; 2.0458/2 ; 2.0458/4]

[I1xx ; I2xx ; I3xx ; I4xx ; I5xx]

[0 ; 0.0116/2 ; 0.0116/2 ; 0.0116/2 ; 0]

[fec1a ; fec2a ; fec3a ; fec4a ; fec5a]

[0.05 ; 0.05 ; 0.05 ; 0.05 ; 0.05]

[fec1b ; fec2b ; fec3b ; fec4b ; fec5b]

[0.05 ; 0.05 ; 0.05 ; 0.05 ; 0.05]

[fv1 ; fv2 ; fv3 ; fv4 ; fv5]

[0.025 ; 0.025 ; 0.025 ; 0.025 ; 0.025]

[gx ; gy ; gz]

[0 ; 0 ; 9.81]

OK Cancel Help Apply

FIG. 19

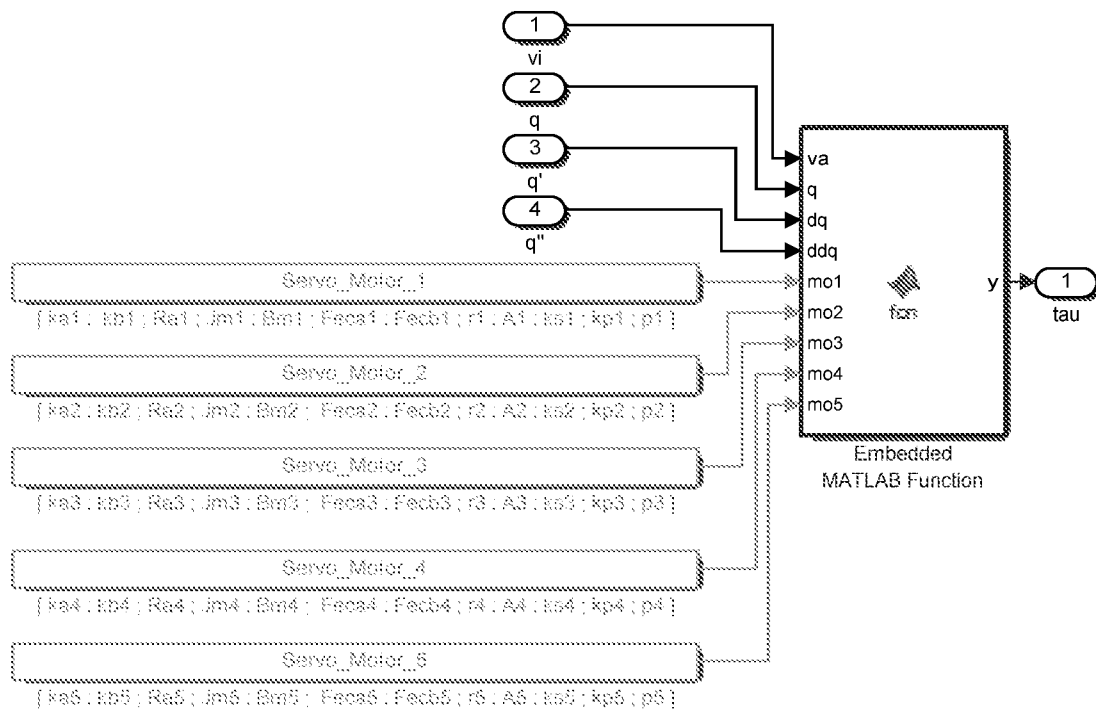


FIG. 20

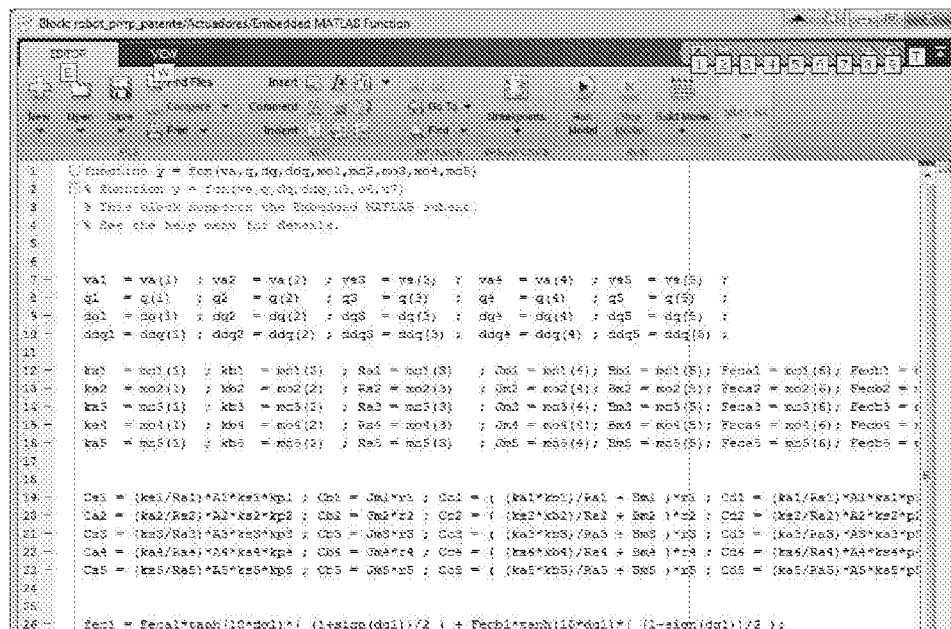
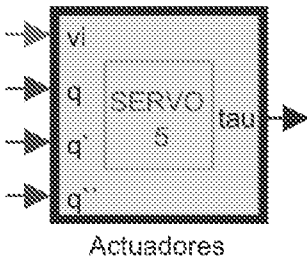
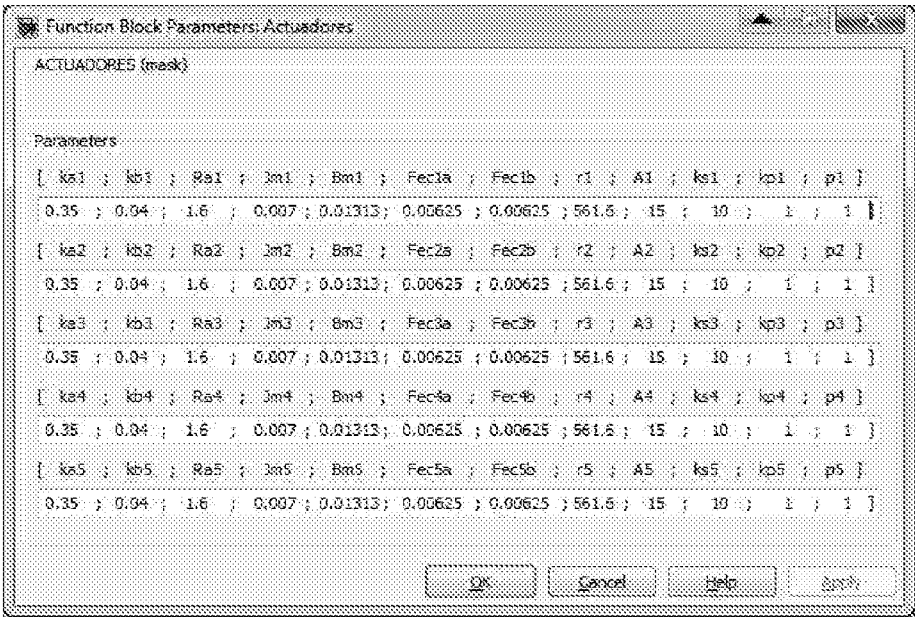


FIG. 21



5

FIG. 22



10

FIG. 23

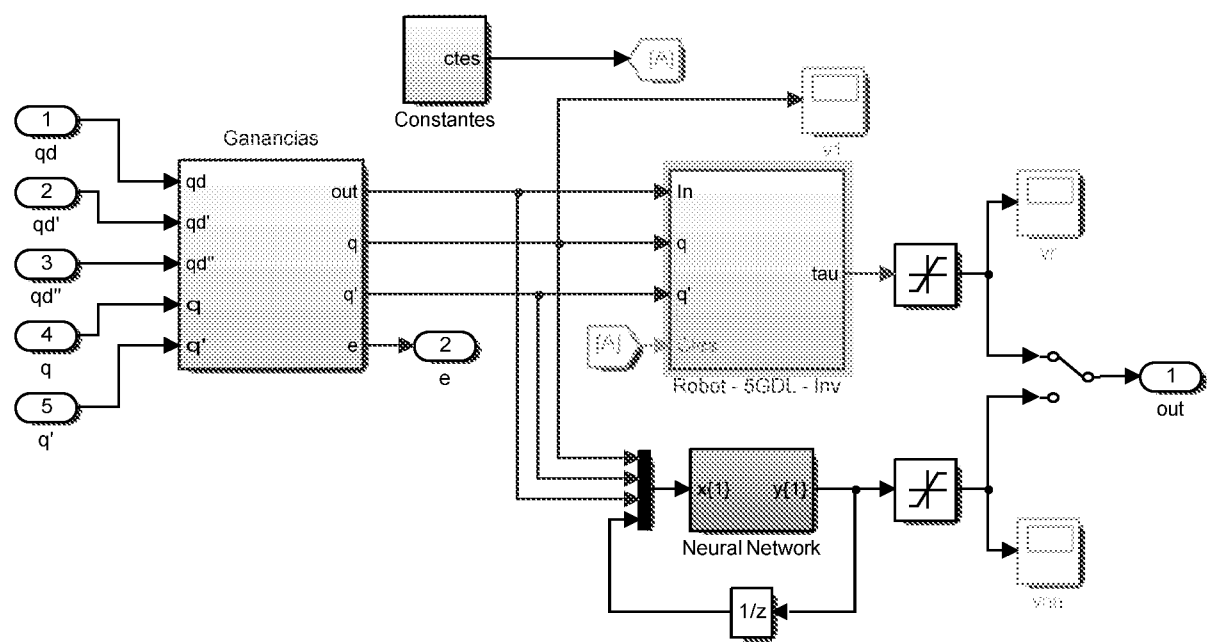


FIG. 24

5

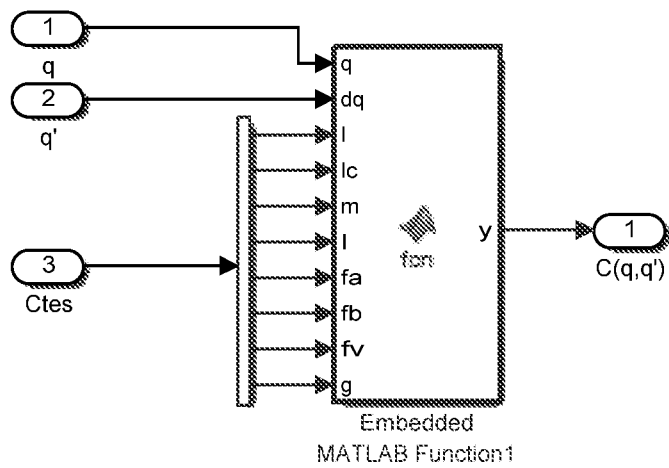


FIG. 25

```

Block: robot_prop_parents/Control/Par Calc - Neuronal Robot - SDC - In/Out q, q' Embedded MATLAB Function1
View
Block: robot_prop_parents/Control/Par Calc - Neuronal Robot - SDC - In/Out q, q' Embedded MATLAB Function1
None Left Top Expand Collapse Expand Collapse
Right Bottom All All
Show function numbers
Enable/disable traceability
Enable/disable code editing

1 function y = Fun(q,dq,l,lc,m,I,fa,fb,fv,q);
2 % This block supports the Embedded MATLAB subset.
3 % See the help menu for details.
4
5
6 q1 = q(1) ; q2 = q(2) ; q3 = q(3) ; q4 = q(4) ; q5 = q(5) ;
7 dq1 = dq(1) ; dq2 = dq(2) ; dq3 = dq(3) ; dq4 = dq(4) ; dq5 = dq(5) ;
8
9 %l01=11/2 ; l02=11/2 ; l03=m1/2 ; I01x=(1/12)*m1*I1^2 ; I02x=(1/12)*m2*I2^2 ; I03x=(1/12)*m3*I3^2 ;
10
11
12 l1 = l(1) ; l2 = l(2) ; l3 = l(3) ; l4 = l(4) ; l5 = l(5) ;
13 lc1 = lc(1) ; lc2 = lc(2) ; lc3 = lc(3) ; lc4 = lc(4) ; lc5 = lc(5) ;
14 m1 = m(1) ; m2 = m(2) ; m3 = m(3) ; m4 = m(4) ; m5 = m(5) ;
15 I1xx = I(1) ; I2xx = I(2) ; I3xx = I(3) ; I4xx = I(4) ; I5xx = I(5) ;
16
17 fec1a = fa(1) ; fec2a = fa(2) ; fec3a = fa(3) ; fec4a = fa(4) ; fec5a = fa(5) ;
18 fec1b = fb(1) ; fec2b = fb(2) ; fec3b = fb(3) ; fec4b = fb(4) ; fec5b = fb(5) ;
19
20 fv1 = fv(1) ; fv2 = fv(2) ; fv3 = fv(3) ; fv4 = fv(4) ; fv5 = fv(5) ;
21
22 qx = q(1) ; qy = q(2) ; qz = q(3) ;
23
24
25 C11 = 0 ;
26 C11 = - 12*(l3*m4*sin(q3) + l3*m5*sin(q3) + l3*m3*sin(q3) + l4*m5*sin(q3 + q4) + l4*m4*sin(q3 + q4))*dq
27 C11 = dq2^2*l3*(l3*m4*sin(q3) + l3*m5*sin(q3) + l3*m3*sin(q3) + l4*m5*sin(q3 + q4) + l4*m4*sin(q3 + q4))

```

FIG. 26

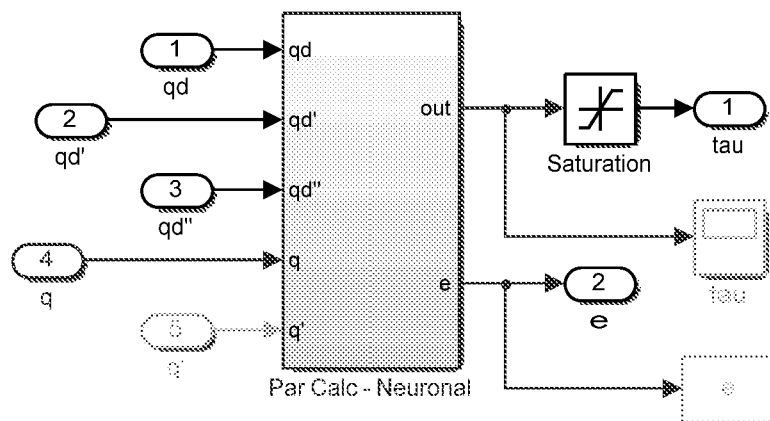


FIG. 27

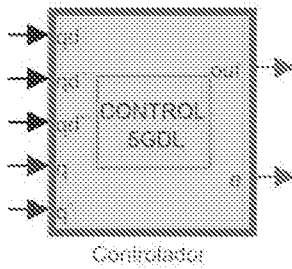
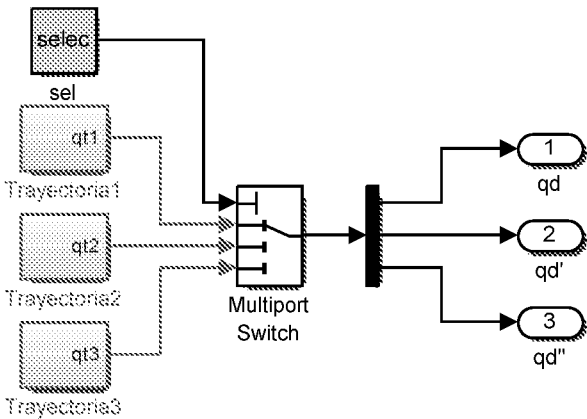
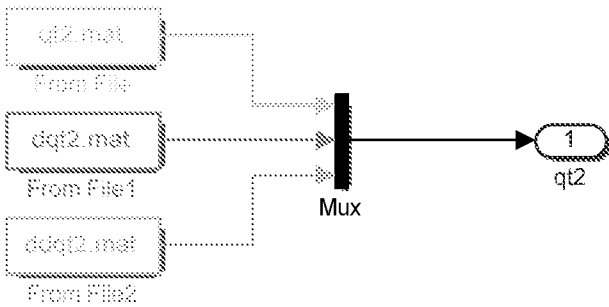


FIG. 28



5

FIG. 29



10

FIG. 30

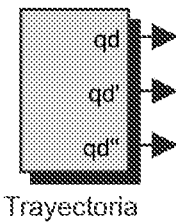


FIG. 31

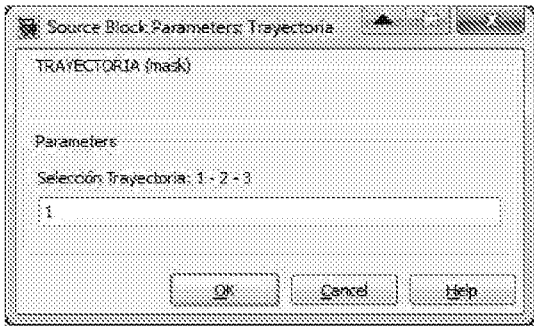


FIG. 32

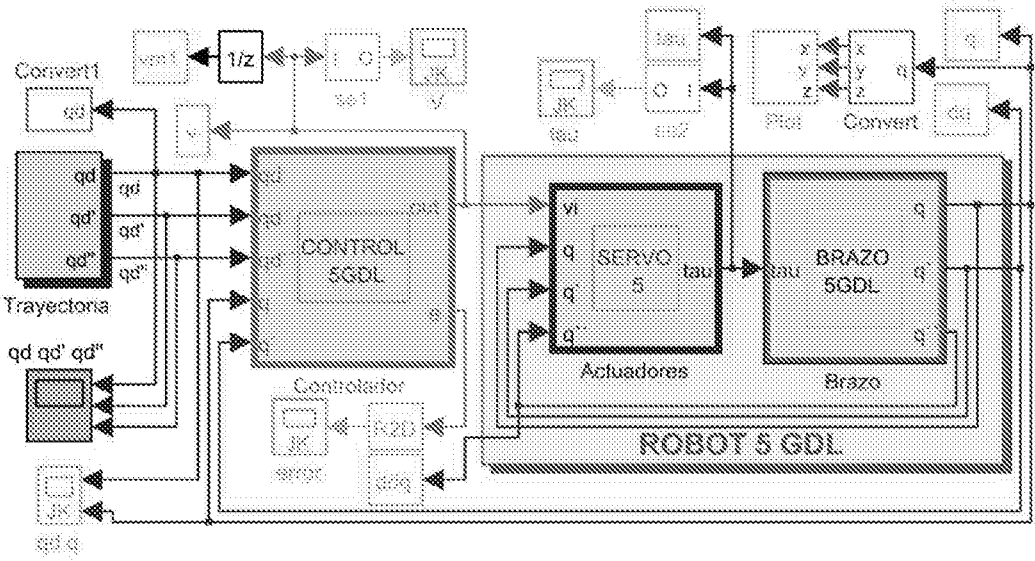
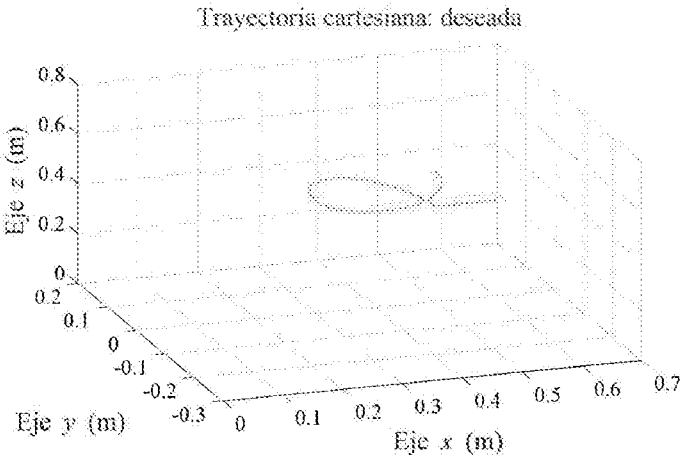
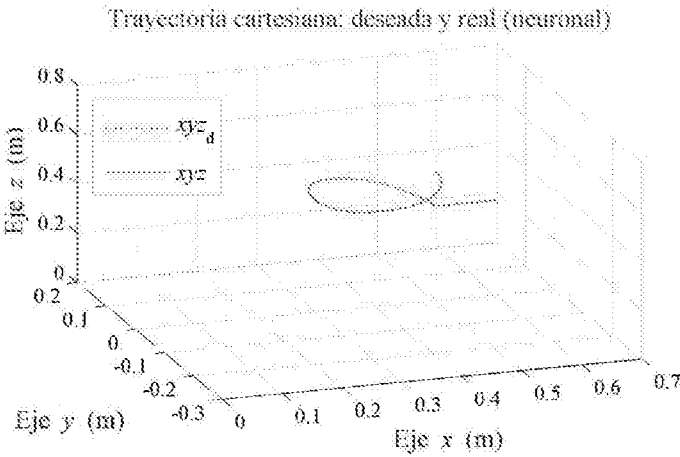


FIG. 33



5

FIG. 34



10

FIG. 35

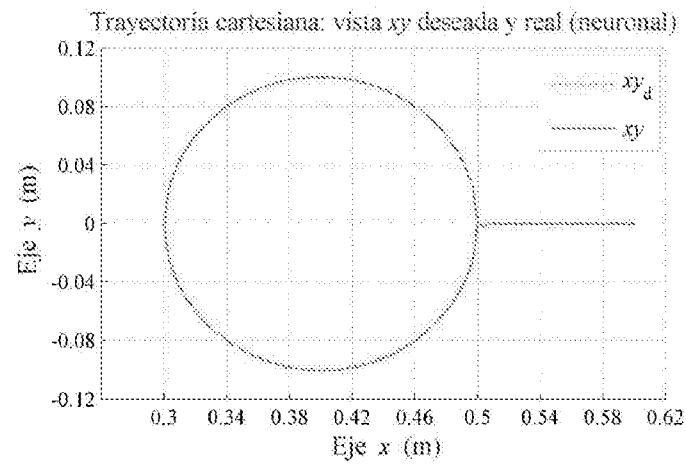


FIG. 36

5

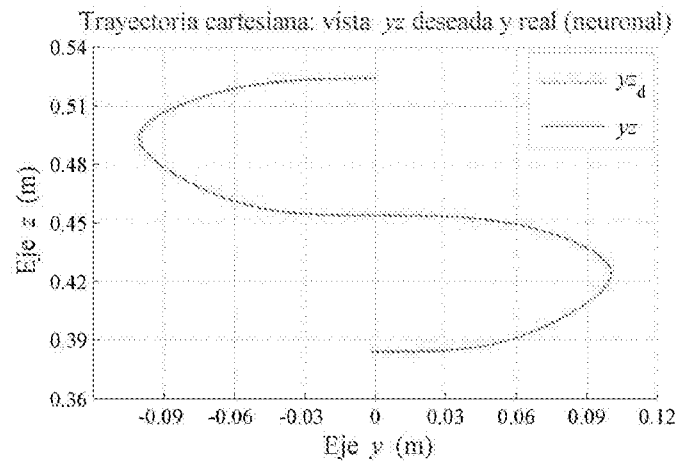


FIG. 37

10

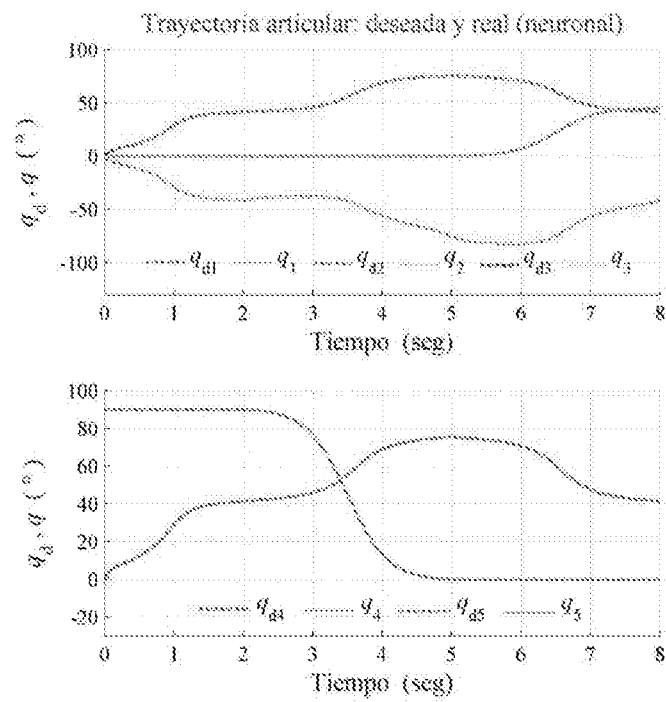


FIG. 38

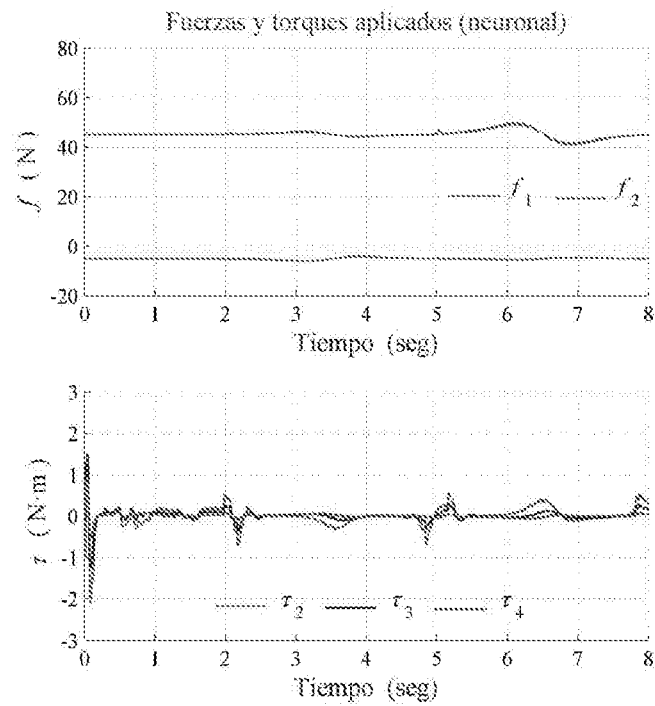


FIG. 39

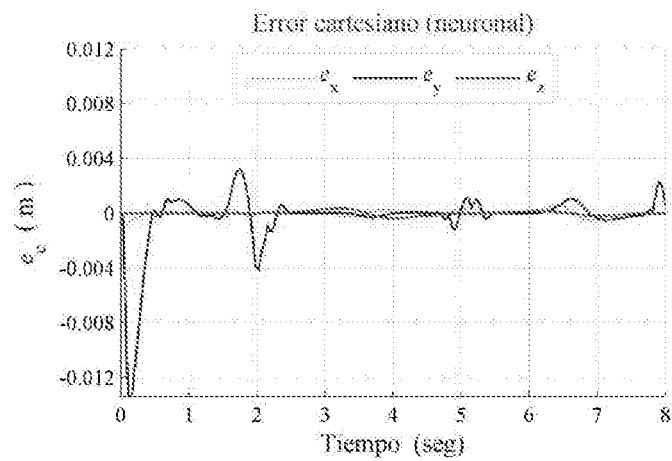


FIG. 40

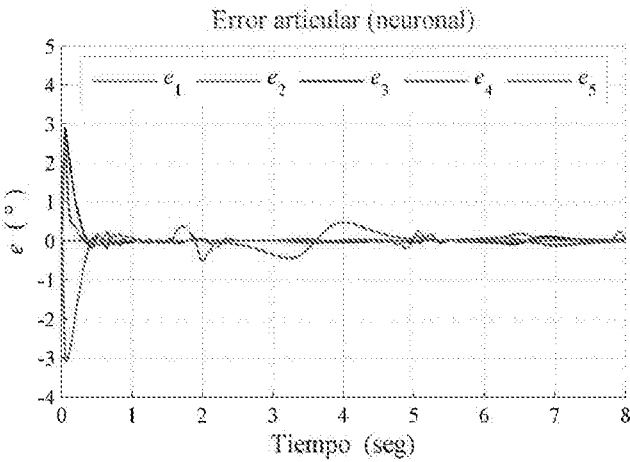


FIG. 41

5

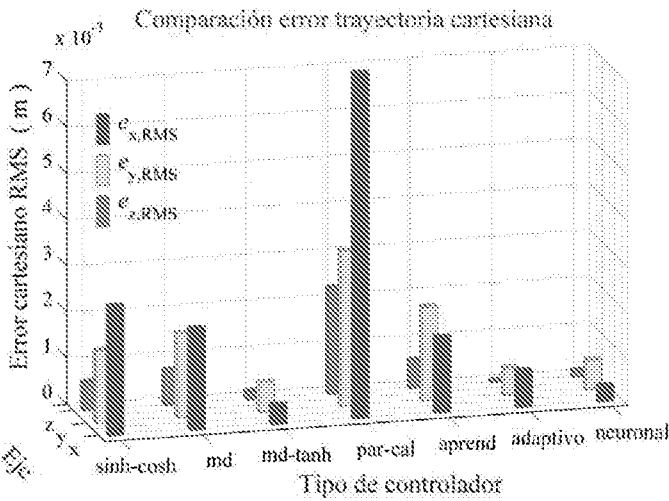


FIG. 42

10

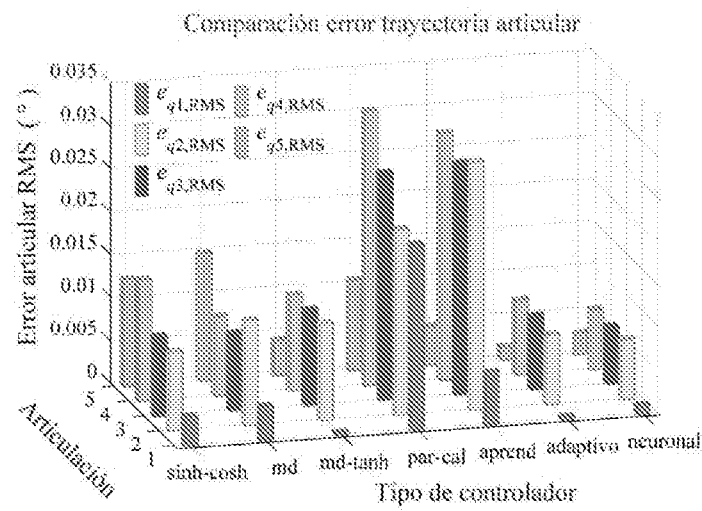


FIG. 43

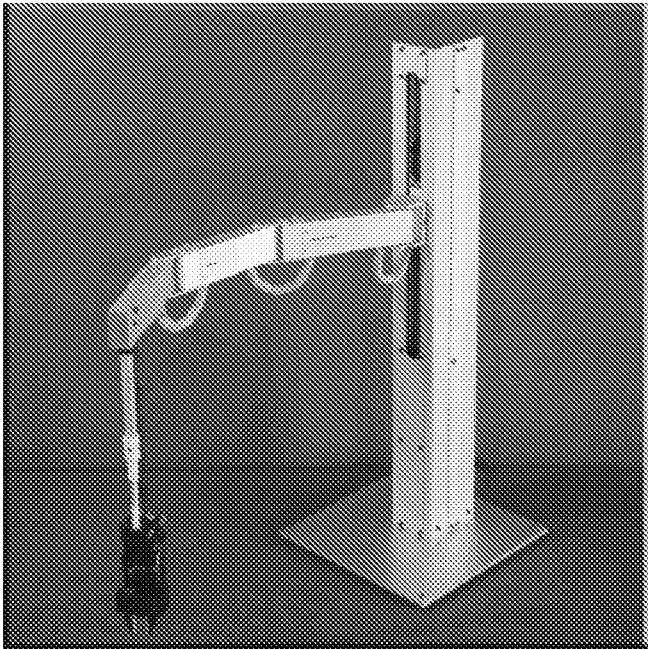


FIG. 44

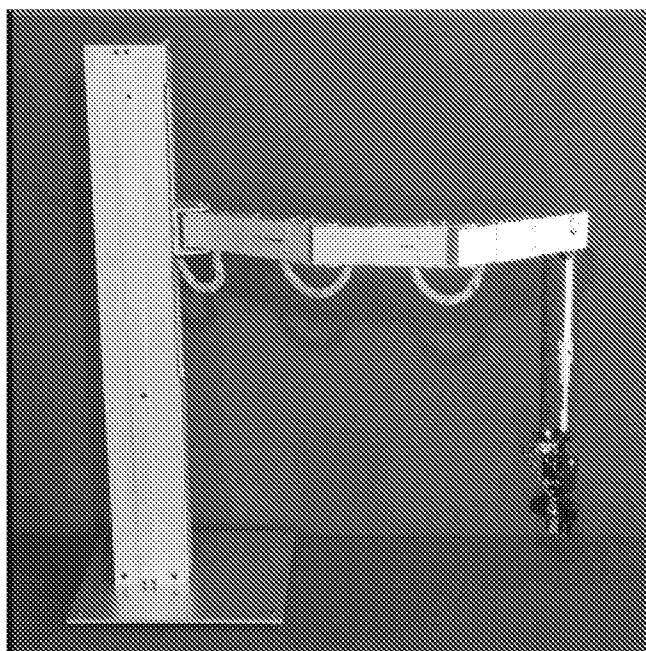


FIG. 45

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CL20 15/050056

A. CLASSIFICATION OF SUBJECT MATTER

(CIP) G09B25/02, B25J9/00 (2016.01)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

(CIP) G09B25/02, B25J9/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the International search (name of data base and, where practicable, search terms used)

EPOQUE, THOMSON, GOOGLE, ESP@CENET, INAPI

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	KERN, J et al. Modelamiento y simulación de un robot redundante de tipo manipulador SCARA. Latin-American Journal of Physics Education (LAJPE). December 2011, Vol. 5, N°4, pages 824-838, ISSN 1870-9095. Fig. 4, 8, 15, 16; Sections B, IV, V, VII, VIII.	1 - 4
Y	URREA C. et al., Modeling, Simulation and Control of a Redundant SCARA-Type Manipulator Robot. International Journal of Advanced Robotic Systems, August 2012, Vol. 9, 58, DOI: 10.5772/51701, ISSN 1729-8806. Fig. 4, 8, 15, 16; Sections 3.2, 4, 5, 7, 8.	1 - 4

☒ Further documents are listed in the continuation of Box C.
 ☒ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the International filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

24/03/2016

Date of mailing of the international search report

13/04/2016

Name and mailing address of the TSA/

INAPI, Av. Libertador Bernardo O'Higgins 194, Piso 17, Santiago, Chile

Facsimile No.

Authorized officer

ARAYA LARA, Hugo Boris

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CL20 15/050056

C [Continuation] DOCUMENTS CONSIDERED TO BE RELEVANT!

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	KERN, J. Desarrollo e implementación de técnicas de control tolerantes a fallas aplicadas en un robot redundante de tipo manipulador SCARA [tesis de grado, en línea]. Santiago. September, 2013 [Recuperated on 2016-03-24]. Recuperated from Internet: < http://repositorio.usach.cl/webclient/DeliveryManager?pid=22983&custom_att_2=simple_viewer&search_terms=robot%20redundante&pds_handle= > Fig. 4.6, 5.9, 5.10, 5.11, 5.25, 5.26, 5.29, 5.73, A.42, A.43; Sections 4.2.8 - 4.2.10, 4.3.4 - 4.3.6, 6.1.	1 - 4
Y	WO 2014/197967 A1 (CAE, INC.); 18 december 2014. Sections 00148, 00156, 00158, 00159, 00164, 00167.	1 - 4
A	URREA, C. et al., Development of a simulation environment applied to the study of fault-tolerant control systems in robotic manipulators. Theoretical and practical comparisons En: Intelligent Mechatronics. Editado por Ganesh Naik. February 2011. ISBN 978-953-307-300-2. Chapter 4. DOI: 10.5772/16181.	

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CL20 15/050056

WO 2014/197967 (A1)	18-12-2014	CA2882009 (A1)	16-04-2015
		CA2887747 (A1)	06-07-2015
		CA2888192 (A1)	26-09-2015
		CA2888879 (A1)	08-07-2015
		CN105102093 (A)	25-1 1-2015
		CN1051 02094 (A)	25-1 1-2015
		CN105247594 (A)	13-06-2016
		EP2939227 (A1)	04-1 1-2015
		EP2941764 (A1)	11-11-2015
		EP2946375 (A1)	25-11-2015
		KR1 0 1552560 (B1)	11-09-2015
		KR101552561 (B1)	11-09-2015
		KR101579145 (B1)	21-12-2015
		TW201 539400 (A)	16-10-2015
		TW201 546770 (A)	16-12-2015
		TW20 1546775 (A)	16-12-2015
		US8843660 (B1)	23-09-2014
		US8868808 (B1)	21-10-2014
		US8990060 (B1)	24-03-2015
		US2015278141 (A1)	01-10-2015
		WO201 5 143528 (A1)	01-10-2015
		WO2015143529 (A1)	01-10-2015
		WO201 5 143531 (A1)	01-10-2015
		WO201 5 143533 (A1)	01-10-2015

INFORME DE BÚSQUEDA INTERNACIONAL

Solicitud internacional N°

PCT/CL20 15/050056

A. CLASIFICACIÓN DEL OBJETO DE LA SOLICITUD

(CIP) G09B25/02, B25J9/00 (2016.01)

De acuerdo con la Clasificación Internacional de Patentes (CIP) o según la clasificación nacional y CIP.

B. SECTORES COMPRENDIDOS POR LA BÚSQUEDA

Documentación mínima buscada (sistema de clasificación seguido de los símbolos de clasificación)

(CIP) G09B25/02, B25J9/00

Otra documentación consultada, además de la documentación mínima, en la medida en que tales documentos formen parte de los sectores comprendidos por la búsqueda

Bases de datos electrónicas consultadas durante la búsqueda internacional (nombre de la base de datos y, si es posible, términos de búsqueda utilizados) EP0QUE, JHOMSON, GOOGLE, ESP@CENET, INAPI

C. DOCUMENTOS CONSIDERADOS RELEVANTES

Categoría*	Documentos citados, con indicación, si procede, de las partes relevantes	Relevante para las reivindicaciones N°
Y	KERN, J et al. Modelamiento y simulación de un robot redundante de tipo manipulador SCARA. Latin-American Journal of Physics Education (LAJPE). Diciembre 2011, Vol. 5, N°4, páginas 824-838, ISSN 1870-9095. Fig. 4, 8, 15, 16; Secciones B, IV, V, VII, VIII.	1 - 4
Y	URREA C. et al., Modeling, Simulation and Control of a Redundant SCARA-Type Manipulator Robot. International Journal of Advanced Robotic Systems, Agosto 2012, Vol. 9, 58, DOI: 10.5772/51701, ISSN 1729-8806. Fig. 4, 8, 15, 16; Secciones 3.2, 4, 5, 7, 8.	1 - 4

☒ En la continuación del Recuadro C se relacionan otros documentos ☒ Los documentos de familias de patentes se indican en el Anexo

* Categorías especiales de documentos citados:	"T"	documento ulterior publicado con posterioridad a la fecha de presentación internacional o de prioridad que no pertenece al estado de la técnica pertinente pero que se cita por permitir la comprensión del principio o teoría que constituye la base de la invención.
"A" documento que define el estado general de la técnica no considerado como particularmente relevante.	"X"	documento particularmente relevante; la invención reivindicada no puede considerarse nueva o que implique una actividad inventiva por referencia al documento aisladamente considerado.
"E" solicitud de patente o patente anterior pero publicada en la fecha de presentación internacional o en fecha posterior.	"Y"	documento particularmente relevante; la invención reivindicada no puede considerarse que implique una actividad inventiva cuando el documento se asocia a otro u otros documentos de la misma naturaleza, cuya combinación resulta evidente para un experto en la materia.
"L" documento que puede plantear dudas sobre una reivindicación de prioridad o que se cita para determinar la fecha de publicación de otra cita o por una razón especial (como la indicada).	"&"	documento que forma parte de la misma familia de patentes.
"O" documento que se refiere a una divulgación oral, a una utilización, a una exposición o a cualquier otro medio.		
"P" documento publicado antes de la fecha de presentación internacional pero con posterioridad a la fecha de prioridad reivindicada.		

Fecha en que se ha concluido efectivamente la búsqueda internacional. 24/03/2016 24/marzo/2016	Fecha de expedición del informe de búsqueda internacional 13/04/2016 13/abril/2016
Nombre y dirección postal de la Administración encargada de la búsqueda internacional INAPI, Av. Libertador Bernardo O'Higgins 194, Piso 17, Santiago, Chile N° de fax	Funcionario autorizado ARAYA LARA, Hugo Boris N° de teléfono 56-2-28870550 56-2-28870551

INFORME DE BÚSQUEDA INTERNACIONAL

Solicitud internacional N°

PCT/CL201 5/050056

C (continuación). DOCUMENTOS CONSIDERADOS RELEVANTES		
Categoría*	Documentos citados, con indicación, si procede, de las partes relevantes	Relevante para las reivindicaciones N°
Y	KERN, J. Desarrollo e implementación de técnicas de control tolerantes a fallas aplicadas en un robot redundante de tipo manipulador SCARA [tesis de grado, en línea]. Santiago. Septiembre, 2013 [recuperado el 2016-03-24]. Recuperado de Internet: < http://repositorio.usach.cl/webclient/DeliveryManager?pid=22983&custom_att_2=simple_viewer&search_terms=robot%20redundante&pds_handle= > Fig. 4.6, 5.9, 5.10, 5.11, 5.25, 5.26, 5.29, 5.73, A.42, A.43; Secciones 4.2.8 - 4.2.10, 4.3.4 - 4.3.6, 6.1 .	1 - 4
Y	WO 2014/197967 A 1 (CAE, INC.); 18 diciembre 2014. Secciones 00148, 00156, 00158, 00159, 00164, 00167.	1 - 4
A	URREA, C. et al., Development of a simulation environment applied to the study of fault-tolerant control systems in robotic manipulators. Theoretical and practical comparisons En: Intelligent Mechatronics. Editado por Ganesh Naik. Febrero 201 1. ISBN 978-953-307-300-2. Capítulo 4. DOI: 10.5772/16181.	

INFORME DE BÚSQUEDA INTERNACIONAL

Información relativa a miembros de familias de patentes

Solicitud internacional N°

PCT/CL20 15/050056

Documento de patente citado en Informe de Búsqueda	Fecha de Publicación	Miembro(s) de Familia	Fecha de Publicación
WO 2014/197967 (A1)	18-12-2014	CA2882009 (A1)	16-04-2015
		CA2887747 (A1)	06-07-2015
		CA2888192 (A1)	26-09-2015
		CA2888879 (A1)	08-07-2015
		CN1051 02093 (A)	25-11-2015
		CN1051 02094 (A)	25-11-2015
		CN105247594 (A)	13-06-2016
		EP2939227 (A1)	04-11-2015
		EP2941764 (A1)	11-11-2015
		EP2946375 (A1)	25-11-2015
		KR1 01552560 (B1)	11-09-2015
		KR101552561 (B1)	11-09-2015
		KR101579145 (B1)	21-12-2015
		TW201 539400 (A)	16-10-2015
		TW201 546770 (A)	16-12-2015
		TW201 546775 (A)	16-12-2015
		US8843660 (B1)	23-09-2014
		US8868808 (B1)	21-10-2014
		US8990060 (B1)	24-03-2015
		US2015278141 (A1)	01-10-2015
		WO201 5 143528 (A1)	01-10-2015
		WO2015143529 (A1)	01-10-2015
		WO201 5 143531 (A1)	01-10-2015
		WO201 5 143533 (A1)	01-10-2015