

Efficient FPGA Hardware Implementation for Robot Manipulator Kinematic Modeling Using Rational Trigonometry

R. Martínez, E. Zamora, *Member, IEEE*, and H. Sossa, *Senior Member, IEEE*

Abstract—A common approach to model kinematics for robot manipulators uses functions such as $\sin()$, $\cos()$ and $\text{atan}()$, however, the rational trigonometry allows to exclude the use of these transcendental functions. Most of the computing processing to control a manipulator is dedicated to solving its kinematics, therefore, reduce the complexity of kinematic equations leads to a more compact and efficient model. In this paper, we propose a geometric analysis to solve the kinematics of robot manipulator of up to three degrees of freedom, avoiding the use of transcendental functions in the kinematic equations. We present three kinematic models for the mechanism positions and their implementations in an FPGA. The experimental results show that our proposed models: 1) use less logic elements compared with an implementation based on parallel CORDIC and 2) has lower latency respect with iterative CORDIC implementation.

Index Terms—FPGA, Kinematics, Rational Trigonometry, Robotics, Transcendental functions.

I. INTRODUCCIÓN

EN las tareas de control y automatización, resolver la cinemática de un manipulador es una tarea fundamental que suele utilizar una gran cantidad de los recursos de computo disponibles, sobre todo en aplicaciones de tiempo real [1]–[3], por tanto, sería útil reducir de alguna forma la complejidad de las ecuaciones cinemáticas. Para mecanismos de pocos grados de libertad (GDL) el modelado cinemático se reduce a un problema geométrico que involucra el uso de funciones trigonométricas, también llamadas funciones trascendentes.

Existen enfoques clásicos que utilizan herramientas como el álgebra lineal y las transformaciones homogéneas para realizar el cálculo de la cinemática, hasta técnicas más novedosas, las cuales utilizan álgebras geométricas en cuatro o cinco dimensiones para resolver los mismos problemas. En la bibliografía se presenta al álgebra geométrica como el estado del arte en la materia, Bayro-Corrochano et al. [4] han examinado las aplicaciones del álgebra geométrica en campos como la visión por ordenador y la cinemática de manipuladores, al igual que Dietmar [5] y Perwass [6] en sus libros. Las tecnologías utilizadas para la implementación de este tipo de algoritmos basan su eficiencia en la gran cantidad de recursos de hardware disponibles en las computadoras actuales, por lo que el tamaño de la implementación no presenta ningún problema para ellos.

Otra forma de reducir la complejidad de los modelos cinemáticos es tratar de eliminar las no linealidades del

mecanismo, así como buscar nuevos algoritmos que lleguen al mismo resultado pero con un menor número de operaciones [7]–[9].

Por lo tanto, es natural preguntarse ¿Es posible modelar la cinemática directa e inversa de un manipulador evitando el uso de funciones trascendentes? Para responder a esta pregunta se presenta un nuevo enfoque para modelar la posición de un manipulador utilizando trigonometría racional. La trigonometría racional fue dada a conocer por el profesor Norman Wilberger de la Universidad de New South en Australia [10]. Este tipo de matemáticas contradicen la creencia de que las funciones trigonométricas junto con los ángulos son la mejor alternativa para describir las propiedades de un triángulo. J. P. Almeida utilizó la trigonometría racional para describir la cinemática directa de un manipulador de seis GDL utilizando matrices de rotación en forma racional e implementó su algoritmo en software [11], [12].

Ya que la tecnología de circuitos integrados ofrece una manera de lograr un alto rendimiento con los FPGA [13]–[18], nuestro objetivo es reducir la complejidad de los circuitos en hardware, para calcular la cinemática de los mecanismos propuestos. En contraste con el trabajo de J. P. Almeida, en este artículo se propone una nueva metodología con un enfoque geométrico en forma racional, sin requerir el uso de matrices. Se presenta por primera vez la implementación en hardware para el modelo cinemático directo e inverso para mecanismos de cadena cinemática abierta de uno, dos y tres GDL, adicionalmente se comparan las implementaciones propuestas con las implementaciones clásicas basadas en funciones trascendentes, demostrando un ahorro significativo en los recursos de hardware utilizados para mecanismos de hasta tres GDL. Es importante mencionar que en este trabajo se trata de evitar el uso de funciones trascendentes no porque existan problemas con su implementación en hardware, sino porque las implementaciones existentes requieren un mayor número de recursos de hardware.

II. TRIGONOMETRÍA RACIONAL

La trigonometría racional se basa en la idea de que el álgebra es más básica que el análisis y que las mediciones reales en geometría elemental deben ser de naturaleza cuadrática en lugar de lineal [19]. A diferencia de la trigonometría clásica, la trigonometría racional no requiere el uso de funciones trascendentes. Dos conceptos fundamentales que forman la base de esta teoría son el *quadrance* y el *spread*.

R. Martínez, E. Zamora y H. Sossa pertenecen al Centro de Investigación en Computación del IPN. Av. Juan de Dios Batiz S/N, Col. Nueva industrial Vallejo, Gustavo A. Madero, 07738, Ciudad de México, México. e-mails: a171074@sagitario.cic.ipn.mx, ezamorag@cic.ipn.mx, hsossa@cic.ipn.mx.

El quadrance mide la separación de dos puntos, mientras que el spread mide la separación de dos líneas.

Si bien el conjunto de números con el que se desarrolló originalmente el quadrance y el spread fue el conjunto de los números racionales, el uso de otro conjunto de números (por ejemplo, los números reales) no está restringido.

El quadrance $Q(A_1, A_2)$ entre los puntos $A_1 \equiv [x_1, y_1]$ y $A_2 \equiv [x_2, y_2]$ es el número

$$Q(A_1, A_2) = (x_2 - x_1)^2 + (y_2 - y_1)^2 \quad (1)$$

donde x_1, x_2, y_1 y y_2 son las coordenadas de los puntos A_1 y A_2 respectivamente, que son números racionales o incluso números reales. Al establecer una relación entre la trigonometría clásica y la racional, el quadrance se puede ver como el cuadrado de la distancia euclidiana.

$$Q(A_1, A_2) = d(A_1, A_2)^2 \quad (2)$$

El spread en el vértice A de un triángulo rectángulo \overline{ABC} no nulo con un ángulo recto en el vértice C , y los quadrances Q_1, Q_2 y Q_3 como se muestra en la Figura 1(b) es

$$s = \frac{Q(B, C)}{Q(A, B)} = \frac{Q_A}{Q_C} \quad (3)$$

Si dos spreads son equivalentes a dos ángulos suplementarios, entonces son iguales entre sí, como se observa en la Figura 1(a). Además, a partir de la ecuación (3), podemos definir la siguiente relación

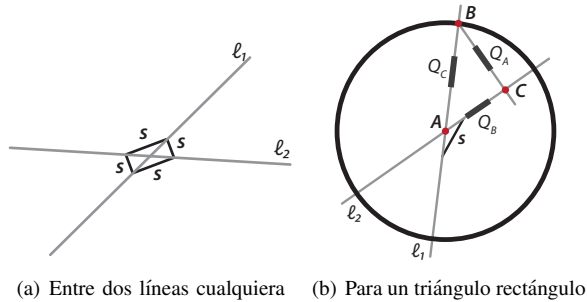


Fig. 1. Definición para el spread.

$$s = \frac{Q(B, C)}{Q(A, B)} = \frac{d(B, C)^2}{d(A, B)^2} = \left(\frac{d(B, C)}{d(A, B)} \right)^2 = \sin^2(\theta) \quad (4)$$

donde θ es un ángulo igual al spread entre Q_B y Q_C .

Además del quadrance y el spread, la trigonometría racional tiene cinco leyes principales. En la siguiente sección se utilizarán varias de estas leyes para formular los modelos cinemáticos que se proponen. Para tener una mejor comprensión de los modelos cinemáticos y las leyes racionales se recomienda consultar las referencias [10] y [20].

III. CINEMÁTICA

En robótica, la cinemática se divide en dos ramas: cinemática directa e inversa. La cinemática directa estudia la posición del mecanismo cuando se asignan los parámetros de la articulación, mientras que la cinemática inversa determina

los parámetros de la articulación cuando el mecanismo está en una posición dada. En esta sección describimos los modelos cinemáticos de acuerdo con el nuevo enfoque propuesto. El modelado se divide en tres partes, comenzando con un mecanismo de un GDL y terminando con un mecanismo de tres GDL. En todas las figuras las articulaciones se representan con un punto gris y el efector final con un punto amarillo, adicionalmente se considera positivas las rotaciones en sentido anti-horario.

A. Modelo de un GDL

La Figura 2(a) muestra el mecanismo de un GDL con el que se realiza el análisis cinemático clásico.

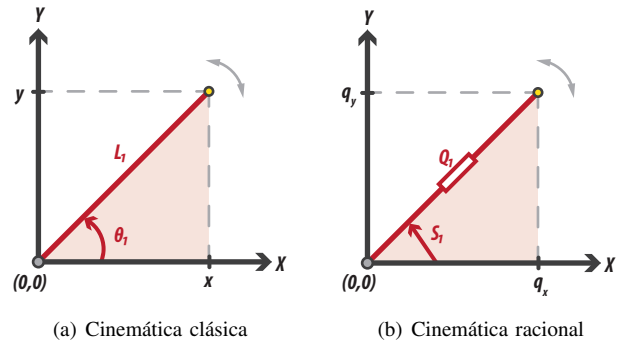


Fig. 2. Configuración para el mecanismo de un GDL.

Para poder modelar este mecanismo con trigonometría racional se tienen que tomar en cuenta algunas consideraciones. La trayectoria de la articulación se divide en cuatro partes iguales porque el spread solo toma valores entre $[0, 1]$. Estas partes pueden empatarse con los cuatro cuadrantes del plano cartesiano. La Figura 2(b) muestra la primera de las cuatro configuraciones posibles del mecanismo de un GDL con el que se realiza un análisis cinemático racional.

Proposición 1. Las ecuaciones que describen la cinética directa de manera racional para el mecanismo de un GDL, están dadas por

$$q_x = Q_1 (1 - s_1) \quad (5)$$

$$q_y = Q_1 s_1 \quad (6)$$

Prueba: Para resolver la cinemática directa ($s_1 \mapsto q_x, q_y$), debemos aplicar la definición de spread (3) de acuerdo con la Figura 2(b). Para obtener la coordenada q_y se tiene que

$$s_1 = \frac{q_y}{Q_1} \quad \rightarrow \quad q_y = Q_1 s_1 \quad (7)$$

En trigonometría, la suma de los ángulos internos de un triángulo es 180° . Dado que un ángulo recto es equivalente a un spread igual a uno, la suma de ambos ángulos agudos es igual a uno; es decir, $s_1 + (1 - s_1) = 1$. Por lo tanto, para obtener el valor de q_x , debemos aplicar la definición de spread (3) a la abertura opuesta del triángulo

$$1 - s_1 = \frac{q_x}{Q_1} \quad \rightarrow \quad q_x = Q_1 (1 - s_1) \quad (8)$$

■

Proposición 2. Las ecuaciones que describen la cinemática inversa en forma racional para el mecanismo de un GDL, están dadas por

$$s_1 = \frac{q_y}{q_x + q_y} \quad (9)$$

Prueba: para resolver la cinemática inversa ($q_x, q_y \mapsto s_1$), tenemos que verificar que Q_1 satisface el teorema de Pitágoras racional, es decir $Q_1 = q_x + q_y$. Aplicando la definición de spread (3), se obtiene s_1 de acuerdo con la siguiente ecuación

$$s_1 = \frac{q_y}{Q_1} = \frac{q_y}{q_x + q_y} \quad (10)$$

donde s_1 representa la separación entre Q_1 y el eje x . ■

B. Modelo de Dos GDL

La Figura 3 muestra el mecanismo de dos GDL con el que se realiza el análisis cinemático clásico. Esta configuración se utiliza para describir un robot SCARA [21].

Para analizar el mecanismo anterior con trigonometría racional, nos basamos en la Figura 4. El recorrido de la segunda articulación se divide en cuatro partes iguales y, por lo tanto, hay cuatro configuraciones diferentes. La implementación para el mecanismo de dos GDL resulta ser más compleja, pero desarrollamos un procedimiento para describir el movimiento en todo el plano cartesiano en la sección 4.

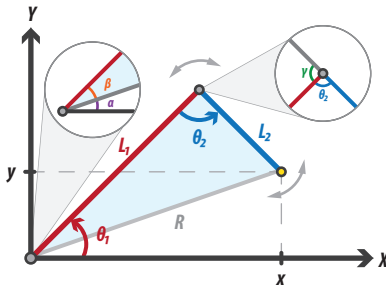


Fig. 3. Configuración clásica para el mecanismo de dos GDL.

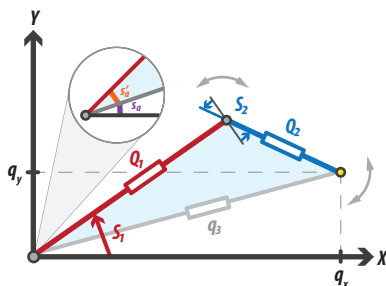


Fig. 4. Configuración racional para el mecanismo de dos GDL. ■

Proposición 3. Las ecuaciones que describen la cinemática directa de forma racional para el mecanismo de dos GDL están dadas por

$$q_x = q_3 (1 - s_a) \quad (11)$$

$$q_y = q_3 s_a \quad (12)$$

Prueba: Para resolver cinemática directa ($s_1, s_2 \mapsto q_x, q_y$), debemos aplicar la ley cruz de acuerdo con la Figura 4

$$q_3 = Q_1 + Q_2 - 2\sqrt{Q_1 Q_2 (1 - s_2)} \quad (13)$$

luego usamos la ley de spread para obtener un valor s'_a

$$\frac{s'_a}{Q_2} = \frac{s_2}{q_3} \quad \longrightarrow \quad s'_a = \frac{s_2 Q_2}{q_3} \quad (14)$$

por último, s_a se calcula utilizando una diferencia de spreads

$$s_a = s_1 \ominus s'_a = \left(\sqrt{s_1(1 - s'_a)} - \sqrt{s'_a(1 - s_1)} \right)^2 \quad (15)$$

utilizando las variables calculadas previamente, las coordenadas del efector final se obtienen de la siguiente forma

$$q_x = q_3 (1 - s_a) \quad (16)$$

$$q_y = q_3 s_a \quad (17)$$

■

Proposición 4. Las ecuaciones que describen la cinemática inversa en forma racional para el mecanismo de dos GDL están dadas por

$$s_1 = s'_a \oplus s_a \quad (18)$$

$$s_2 = 1 - \frac{(Q_1 + Q_2 - q_3)^2}{4 Q_1 Q_2} \quad (19)$$

Prueba: Para resolver la cinemática inversa ($q_x, q_y \mapsto s_1, s_2$), se calcula el valor q_3 utilizando el teorema de Pitágoras. Luego, utilizando la definición de spread (3), calculamos s_a de la siguiente manera

$$s_a = \frac{q_y}{q_3} \quad (20)$$

posteriormente, usando la ley de spread obtenemos la variable s'_a de la misma manera que en la ecuación (14). Aplicando el teorema de Pitágoras para obtener el valor de q_3

$$q_3 = q_x + q_y \quad (21)$$

se puede calcular s_1 aplicando la suma de spreads

$$s_1 = s'_a \oplus s_a = \left(\sqrt{s'_a(1 - s_a)} + \sqrt{s_a(1 - s'_a)} \right)^2 \quad (22)$$

finalmente, aplicando la ley cruz se obtiene la variable s_2

$$s_2 = 1 - \frac{(Q_1 + Q_2 - q_3)^2}{4 Q_1 Q_2} \quad (23)$$

■

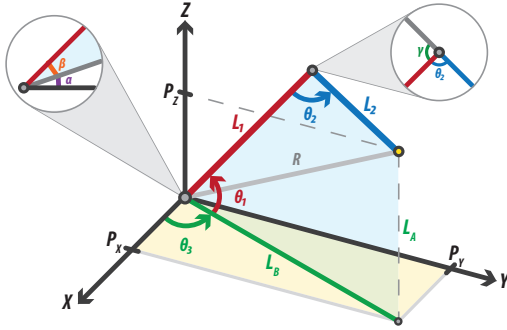


Fig. 5. Configuración clásica para el mecanismo de tres GDL.

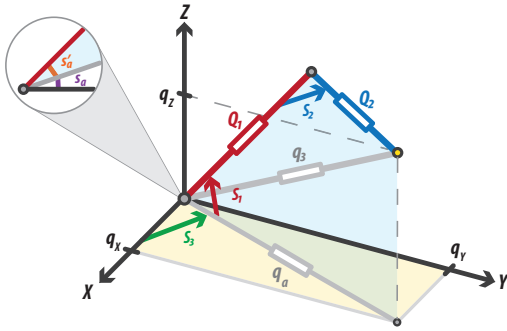


Fig. 6. Configuración racional para el mecanismo de tres GDL.

C. Modelo de Tres GDL

La Figura 5 muestra el mecanismo de tres GDL con el que se realiza el análisis cinemático clásico. A diferencia de los modelos anteriores donde los ejes de rotación de las articulaciones eran perpendiculares al plano xy , este mecanismo se puede mover en el espacio tridimensional. Este tipo de mecanismo es utilizado para describir un robot esférico [22].

Ahora analizamos el mecanismo de tres GDL de forma racional. Después de dividir el recorrido de las articulaciones en cuatro partes iguales, hay 64 regiones posibles del espacio de trabajo. La Sección 4 desarrolla en detalle el procedimiento para describir el movimiento en todo el espacio tridimensional. **Proposición 5.** Las ecuaciones que describen la cinemática directa en forma racional para el mecanismos de tres GDL están dadas por

$$q_x = q_a - q_y \quad (24)$$

$$q_y = s_3 q_a \quad (25)$$

$$q_z = s_a q_3 \quad (26)$$

Prueba: Para resolver la cinemática directa ($s_1, s_2, s_3 \mapsto q_x, q_y, q_z$), se tienen que calcular las variables q_3 y s'_a de la misma forma que en las ecuaciones (13) y (14). Se obtiene el valor de q_a usando el teorema de Pitágoras

$$q_a = q_3 - q_z \quad (27)$$

y el valor de s_a aplicando una diferencia de spreads

$$s_a = s_1 \ominus s'_a = \left(\sqrt{s_1(1-s'_a)} - \sqrt{s'_a(1-s_1)} \right)^2 \quad (28)$$

luego podemos calcular el valor de las coordenadas deseadas, para q_y y q_z aplicando la definición de spread (3)

$$s_3 = \frac{q_y}{q_a} \rightarrow q_y = s_3 q_a \quad (29)$$

$$s_a = \frac{q_z}{q_3} \rightarrow q_z = s_a q_3 \quad (30)$$

finalmente usando el teorema de Pitágoras obtenemos q_x

$$q_x = q_a - q_y \quad (31)$$

Proposición 6. Las ecuaciones que describen la cinemática inversa en forma racional para el mecanismo de tres GDL están dadas por

$$s_1 = s_a \oplus s'_a \quad (32)$$

$$s_2 = 1 - \frac{(Q_1 + Q_2 - q_3)^2}{4Q_1Q_2} \quad (33)$$

$$s_3 = \frac{q_y}{q_a} \quad (34)$$

Prueba: Para resolver cinemática inversa ($(q_x, q_y, q_z \mapsto s_1, s_2, s_3)$), encontramos el valor de q_a y q_3 aplicando el teorema de Pitágoras

$$q_a = q_y + q_x \quad (35)$$

$$q_3 = q_a + q_z \quad (36)$$

el valor de s_a se calcula utilizando la definición de spread (3)

$$s_a = \frac{q_z}{q_3} \quad (37)$$

y el valor de s'_a se obtienen aplicando la ley de spread de la misma manera que en (14). El primer spread que buscamos está definido por una suma de spreads

$$s_1 = s_a \oplus s'_a = \left(\sqrt{s_a(1-s'_a)} + \sqrt{s'_a(1-s_a)} \right)^2 \quad (38)$$

el segundo spread s_2 se calcula usando la ley cruz

$$s_2 = 1 - \frac{(Q_1 + Q_2 - q_3)^2}{4Q_1Q_2} \quad (39)$$

y finalmente s_3 se obtiene usando la definición de spread (3)

$$s_3 = \frac{q_y}{q_a} \quad (40)$$

D. Aplicación del Modelo de Tres GDL

Antes de implementar los modelos cinemáticos en un FPGA, decidimos verificar las ecuaciones en el software *Geogebra* para corroborar su correcto funcionamiento. Las simulaciones nos dieron una guía de cómo funcionan esos modelos. Aquí, damos un ejemplo para calcular la cinemática directa del mecanismo de tres GDL aplicando el procedimiento descrito en la sección anterior.

Supongamos que las dimensiones de los eslabones están dadas por los cuadrantes $Q_1 = 9$ y $Q_2 = 4$, adicionalmente las aberturas de las articulaciones están dadas por los spreads $s_1 = 0.5$, $s_2 = 0.6$ y $s_3 = 0.3$ de acuerdo con la Figura 6. Encuentre la posición del mecanismo para el primer y tercer

octantes cuando la segunda articulación está en la primera configuración.

El primer paso es calcular las variables auxiliares.

$$q_3 = 9 + 4 - 2\sqrt{9 \cdot 4 \cdot (1 - 0.6)} = 5.41 \quad (41)$$

$$s'_a = \frac{(0.6) \cdot 4}{5.4105} = 0.44 \quad (42)$$

$$s_a = \left(\sqrt{0.2783} - \sqrt{0.2218} \right)^2 = 3.2 \times 10^{-3} \quad (43)$$

$$q_a = (5.4105) - (0.017327) = 5.39 \quad (44)$$

Ahora, al usar las ecuaciones (24), (25) y (26) obtenemos la coordenadas deseadas de forma racional

$$q_x = (5.39) \cdot (1 - 0.3) = 3.7748 \quad (45)$$

$$q_y = (5.39) \cdot (0.3) = 1.6178 \quad (46)$$

$$q_z = (5.39) \cdot (3.2 \times 10^{-3}) = 0.0173 \quad (47)$$

para obtener el valor real de las coordenadas aplicamos raíz cuadrada a cada una de las ecuaciones anteriores

$$x = \sqrt{q_x} = 1.942 \quad (48)$$

$$y = \sqrt{q_y} = 1.281 \quad (49)$$

$$z = \sqrt{q_z} = 0.132 \quad (50)$$

Este resultado solo es válido para el primer octante. Para encontrar las coordenadas en el tercer octante, tenemos que cambiar los signos de las coordenadas x y y . La Figura 7 muestra las regiones para el primer y tercer octante, así como la configuración del mecanismo para este ejemplo.

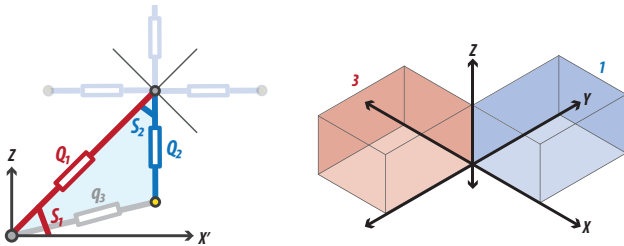


Fig. 7. Configuración del mecanismo para el ejemplo anterior.

IV. IMPLEMENTACIÓN DE MODELOS

En esta sección se describe cómo implementar en hardware las ecuaciones obtenidas en la sección anterior, todas las ecuaciones fueron sintetizadas en un FPGA de la familia *Cyclone V* utilizando lenguaje VHDL [20]. Ya que las ecuaciones solo funcionan en una región acotada del espacio, a continuación se explicará como se puede mover el mecanismo de forma continua en todo su espacio de trabajo.

A. Implementación para el Mecanismo de un GDL

Para implementar las ecuaciones que describen la cinemática directa del mecanismo de un GDL, se toman las ecuaciones (5) y (6). Sin embargo, para mostrar los valores

reales de las coordenadas cartesianas x y y , se debe aplicar una raíz cuadrada a cada una de las coordenadas

$$x = \sqrt{Q_1(1 - s_1)} \quad (51)$$

$$y = \sqrt{Q_1 s_1} \quad (52)$$

El spread toma valores en el intervalo $[0, 1]$ cuyo equivalente en grados es $[0^\circ, 90^\circ]$. Para obtener una descripción del mecanismo en los cuatro cuadrantes, debemos cambiar los signos de las coordenadas x y y . Para la implementación en hardware, se utiliza un selector con un rango entre $[0, 4]$, este selector es responsable de cambiar los signos de las coordenadas de acuerdo con la Tabla I.

TABLA I
SIGNOS DE LAS COORDENADAS POR CUADRANTE

Cuadrante	s_1	$signo(x)$	$signo(y)$
1	$[0, 1]$	\oplus	\oplus
2	$[1, 2]$	\ominus	\oplus
3	$[2, 3]$	\ominus	\ominus
4	$[3, 4]$	\oplus	\ominus

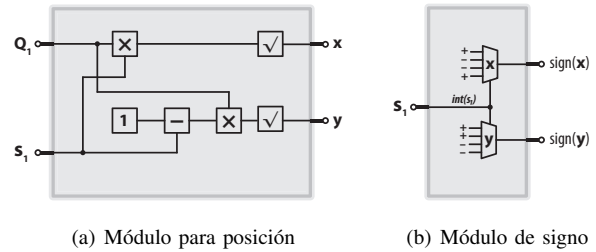


Fig. 8. Diagrama a bloques para el mecanismo de un GDL.

Los diagramas a bloques mostrados en la Figura 8 describen la implementación en hardware para el mecanismo de un GDL. El circuito de la Figura 8(a) calcula la posición del efector final utilizando las ecuaciones (51) y (52), mientras que el circuito de la Figura 8(b) realiza el cambio de signo tomando en cuenta la parte entera del selector s_1 de acuerdo con la Tabla I.

B. Implementación para el Mecanismo de Dos GDL

El modelo para el mecanismo de dos GDL se basa en la implementación anterior. Las operaciones son las mismas pero tenemos que comparar algunas señales antes de asignar las variables. La forma general para calcular las coordenadas del efector final es la siguiente

$$x_{2i} = \sqrt{q_{3i}(1 - s_{xi})} \quad (53)$$

$$y_{2i} = \pm \sqrt{q_{3i} s_{xi}} \quad (54)$$

El signo de y_{2i} está sujeto a la condición $C_i \geq C'_i$, donde C_i y C'_i son las condiciones para que el segundo eslabón pertenezca a i -ésimo cuadrante. En otras palabras, si esta condición es verdadera, entonces asignamos a y_{2i} el valor $\sqrt{q_{3i}s_{xi}}$ de lo contrario $-\sqrt{q_{3i}s_{xi}}$, donde $i = 1, 2, 3, 4$ representa el cuadrante donde se encuentra el segundo eslabón, de acuerdo con el valor de s_2 .

Los valores de C_i y C'_i son diferentes para cada cuadrante, por lo tanto, estas variables cambian dependiendo de dónde se encuentre el eslabón, sin embargo, la estructura de las operaciones no cambia. La Tabla II muestra las operaciones para calcular las coordenadas del efector final en el primer cuadrante del plano cartesiano. La tercera columna de la tabla corresponde a la condición $C_i \geq C'_i$.

TABLA II
VALOR PARA LAS COORDENADAS DEL MODELO DE DOS GDL

Cuadrante	s_2	Valor de coordenadas	Condición
1	[0, 1]	$x_{2,1} = \sqrt{q'_3(1 - s_b)}$ $y_{2,1} = \pm\sqrt{q'_3 s_b}$	$h \geq h'$
2	[1, 2]	$x_{2,2} = \sqrt{q_3(1 - s_a)}$ $y_{2,2} = \pm\sqrt{q_3 s_a}$	$s_1 \geq s'_a$
3	[2, 3]	$y_{2,3} = \sqrt{q_3(1 - s''_a)}$ $x_{2,3} = \pm\sqrt{q_3 s''_a}$	$(1 - s_1) \geq s'_a$
4	[3, 4]	$y_{2,4} = \sqrt{q'_3(1 - s''_b)}$ $x_{2,4} = \pm\sqrt{q'_3 s''_b}$	$h \geq h'$

Los valores h y h' señalan si el efector final está en el mismo cuadrante que el primer eslabón. El valor $h = s_1 Q_1$ representa la altura alcanzada por el primer eslabón Q_1 y $h' = (1 - (s_2 \ominus (1 - s_1))^2) Q_2$ es la altura que va desde la segunda articulación hasta el efector final del mecanismo con respecto al eje y . Cuando la condición $h \geq h'$ no es verdadera, esto implica que el efector final del mecanismo está fuera del cuadrante especificado por el selector s_1 .

Las variables involucradas en los cálculos cinemáticos de acuerdo con la Tabla II son

$$s_a = s_1 \ominus s'_a \tag{55}$$

$$s'_a = \frac{s_2 Q_2}{q_3} \tag{56}$$

$$s''_a = (1 - s_1) \ominus s'_a \tag{57}$$

$$s_b = s_1 \ominus s'_b \tag{58}$$

$$s'_b = \frac{s_2 Q_2}{q'_3} \tag{59}$$

$$s''_b = (1 - s_1) \ominus s'_b \tag{60}$$

las configuraciones para los cuadrantes uno y dos difieren de los cuadrantes tres y cuatro porque las coordenadas x se intercambian por las coordenadas y . Para definir la posición del efector final en cualquier parte del plano cartesiano, asignamos los signos a las coordenadas x_{2i} y y_{2i} de acuerdo con la Tabla I. Estos signos se basan en el parámetro s_1 correspondiente al spread de la primera articulación.

La Figura 9 muestra el diagrama a bloques de la implementación en hardware para las ecuaciones cinemáticas del mecanismo de dos GDL. La primera parte del circuito se encarga de calcular las variables auxiliares que están en

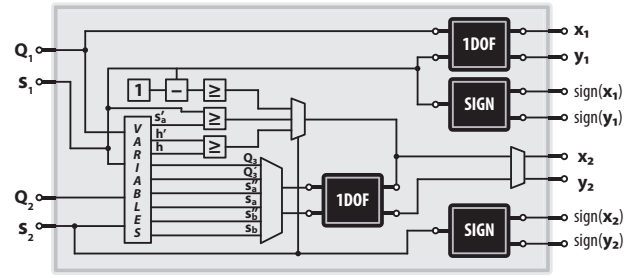


Fig. 9. Diagrama a bloques para el mecanismo de dos GDL.

función de los cuatro parámetros de entrada de acuerdo con la Tabla II, mientras que los módulos *1DOF* corresponden al circuito de la Figura 8(a) y los módulos *SIGN* corresponden a la Figura 8(b). La complejidad de este circuito aumenta porque ahora se tienen que calcular las cuatro variables involucradas en las comparaciones, de acuerdo con la tercera columna de la Tabla II.

C. Implementación para el Mecanismo de Tres GDL

La implementación del mecanismo de tres GDL se basa en las dos implementaciones anteriores. Para especificar cualquier punto en el espacio tridimensional, necesitamos al menos tres parámetros; por lo tanto, usamos un módulo de un GDL y luego un módulo de dos GDL para describir la posición del nuevo mecanismo. En este caso, el circuito de *2DOF* corresponde al circuito descrito en la Figura 9 y se encarga de calcular las coordenadas x y z ; mientras que los circuitos *1DOF* y *SIGN* corresponden a los circuitos mostrados en la Figura 8 y se encargan de calcular la magnitud y el signo de la coordenada y respectivamente.

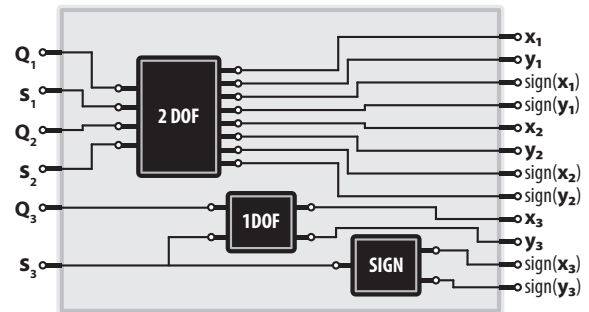


Fig. 10. Diagrama a bloques para el mecanismo de tres GDL.

En aplicaciones reales, un robot esférico no funciona en los ocho octantes del espacio de trabajo. En consecuencia, para este experimento solo analizamos los cuatro octantes superiores del espacio de trabajo. Los selectores utilizados para las articulaciones del mecanismo de tres GDL tienen los siguientes rangos: para la primera articulación $s_1 = [0, 2]$, para la segunda articulación $s_2 = [0, 4]$ y para la última articulación $s_3 = [0, 4]$. La Figura 10 muestra el diagrama a bloques de la implementación en hardware de las ecuaciones cinemáticas para el mecanismo de tres GDL.

En la mayoría de los robots industriales es habitual que los primeros tres GDL se encarguen de posicionar la herramienta y los últimos tres a orientarla. Los seis modelos propuestos anteriormente así como las implementaciones en hardware permiten modelar no solo los tres primeros GDL, de acuerdo con la Figura 11 sería posible modelar hasta la quinta articulación de un robot de seis GDL ó hasta la cuarta articulación en el caso de un robot de cinco GDL (omitiendo la articulación 4 de la Figura 11). Esto es posible debido a que la posición espacial de las articulaciones cuatro y cinco es independiente de su posición articular.

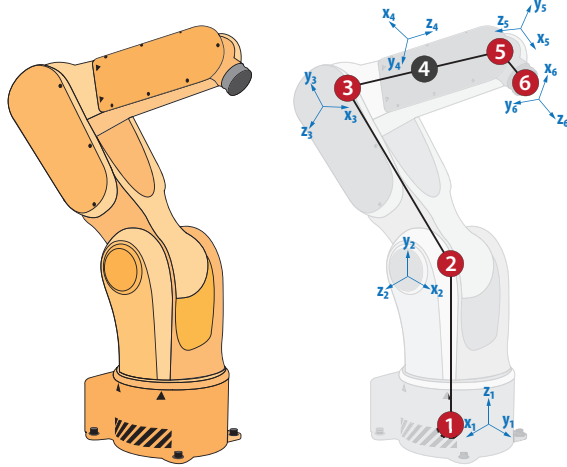


Fig. 11. Estructura de robot industrial.

D. Generalización para el Modelado de Mecanismos

Al tratar de generalizar la metodología propuesta en la sección anterior a más de tres GDL, se observó que conforme crecía el número de GDL los modelos resultaban más complicados y el ahorro de hardware desaparecía, por ello se decidió realizar una nueva implementación que tuviera la capacidad de poder modelar mecanismos que tuvieran un número arbitrario de GDL. Basado en el modelo de tramas fijas (MFF) [11], [12] es posible describir un mecanismo de n articulaciones rotacionales desarrollando la multiplicación $P = L \times R \times B$.

$$L = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \quad R = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & -1 & 1 \\ 1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & -1 \\ 1 & -1 & -1 \end{bmatrix}$$

$$B = \begin{bmatrix} \sqrt{s_\theta s_\varphi R} & 0 & 0 \\ 0 & \sqrt{(1-s_\theta)s_\varphi R} & 0 \\ 0 & 0 & \sqrt{(1-s_\varphi)R} \end{bmatrix}$$

La multiplicación de las matrices L y R produce una matriz de dimensión $[n \times 3]$ la cual debe multiplicarse por la matriz B . Cada renglón de la matriz L representa una articulación, donde solo un elemento toma el valor de 1. La matriz B es

una matriz diagonal donde R es cuadrante de cada eslabón, así como s_θ y s_φ son los spreads de cada uno de los eslabones referentes la trama en la que están. Los términos de la matriz B se calculan con el módulo de la Figura 12, que es similar al módulo 1DOF visto anteriormente.

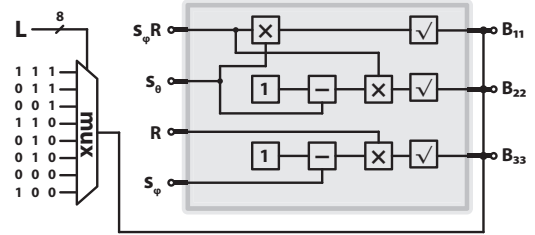


Fig. 12. Módulo para la implementación del método MFF

El método de MFF no ofrece una ventaja en cuanto al ahorro de hardware para mecanismos de pocos GDL debido a las operaciones adicionales que requiere, por ejemplo, la selección de los renglones de la matriz L y las sumas parciales para obtener el valor de la n -ésima articulación. La ventaja que tiene esta implementación es la escalabilidad que posee, ya que para modelar mecanismos con n GDL solo se requiere el uso recurrente de n módulos como el de la Figura 12.

V. RESULTADOS

Los modelos obtenidos en la sección anterior se sintetizaron en un FPGA para encontrar el número de elementos lógicos utilizados. El dispositivo utilizado fue un FPGA de la familia Cyclone V de Intel (antes Altera) y el software utilizado fue Quartus Prime versión 16.0. También implementamos en un FPGA las ecuaciones cinemáticas clásicas para modelar los mecanismos utilizando el algoritmo CORDIC [23]–[26].

El algoritmo CORDIC nos permitió calcular los valores de las funciones trascendentes involucradas en la cinemática directa e inversa para cada uno de los mecanismos. La implementación del algoritmo de CORDIC fue realizada con una aritmética de punto fijo con signo de 16 bits, 1 bit de signo, 7 bits para la parte entera y 8 bits para la parte fraccionaria. La Tabla III muestra el tamaño de la implementación para todos los modelos cinemáticos directos. Los elementos lógicos son una unidad de medida básica que tienen los FPGA para sintetizar los circuitos.

TABLA III
RECURSOS DE HARDWARE PARA LA CINEMÁTICA DIRECTA

GDL	Trigonometría	EL
1	Clásica	1573
	Racional	359
2	Clásica	1657
	Racional	582
3	Clásica	1796
	Racional	918

El porcentaje de ahorro de elementos lógicos para cinemática directa es 77% para el mecanismo de un GDL, 64% para el de dos GDL y 48% para tres GDL. Este ahorro es debido a que la implementación paralela de CORDIC necesita al menos 1400 elementos lógicos y también se necesita implementar una unidad de control para gestionar la entrada de datos. Más grados de libertad implica una unidad de control más compleja y, por tanto, más elementos lógicos. La Tabla IV muestra el tamaño de la implementación para todos los modelos de cinemática inversa.

TABLA IV
RECURSOS DE HARDWARE PARA LA CINEMÁTICA INVERSA

<i>GDL</i>	<i>Trigonometría</i>	<i>EL</i>
1	Clásica	1569
	Racional	317
2	Clásica	1673
	Racional	671
3	Clásica	1787
	Racional	1023

El porcentaje de ahorro de elementos lógicos para cinemática inversa es del 79% para el mecanismo de un GDL, del 59% para el de dos GDL y del 42% para el de tres GDL.

Las tablas anteriores muestran que el ahorro de elementos lógicos es notable. La mayoría de los elementos lógicos utilizados en la implementación racional fueron utilizados para el cálculo de la raíz cuadrada (102 EL) [27]. En contraste, la implementación paralela del algoritmo CORDIC utiliza una gran cantidad de elementos lógicos en comparación con la implementación iterativa, segmentada o híbrida. Esto es porque el circuito es puramente combinatorio. El algoritmo CORDIC iterativo requiere solo una etapa de procesamiento a diferencia del CORDIC paralelo, sin embargo, esta etapa necesita un uso recursivo de al menos 11 iteraciones para calcular el resultado [28]–[30].

La implementación del algoritmo CORDIC paralelo solo requiere circuitos combinatorios, pero es necesario almacenar el resultado de las operaciones parciales para poder calcular el resultado final. Por tanto, hay que realizar un circuito secuencial que controle el flujo de datos (máquina de estado). Pasa lo mismo con la implementación basada en trigonometría racional. La Tabla V muestra la cantidad de ciclos de reloj que toma cada algoritmo para calcular un resultado. Esta comparación se realizó entre nuestras tres implementaciones de hardware y CORDIC iterativo a una frecuencia de 50 MHz. El número de ciclos de reloj se denomina latencia del circuito y es un parámetro importante que debe tenerse en cuenta al diseñar circuitos digitales.

CORDIC iterativo disminuye el área en el FPGA, pero el costo a pagar es un circuito más complejo de controlar y un aumento en la latencia del circuito. Como hemos mostrado en la Tabla V, la latencia de los circuitos basados en CORDIC iterativo aumenta dramáticamente con los GDL, por cada vez que calculamos una función trigonométrica con CORDIC, la

TABLA V
LATENCIA EN CICLOS DE RELOJ PARA CADA ALGORITMO

<i>GDL</i>	<i>Trigonometría</i>	<i>Ciclos de Reloj</i>
1	Clásica	22
	Racional	2
2	Clásica	25
	Racional	10
3	Clásica	52
	Racional	15

latencia del circuito aumenta al menos 11 ciclos de reloj más. Una ventaja de CORDIC es que puede calcular divisiones y raíces cuadradas, sin embargo, si queremos calcular la raíz cuadrada con CORDIC necesitamos aplicar una suma y una resta como procesamiento previo y agregar al menos 11 ciclos de reloj más para cada operación.

VI. CONCLUSIONES

La principal contribución de nuestra investigación fue la propuesta de seis modelos matemáticos para el cálculo de la cinemática directa e inversa de mecanismos de hasta tres GDL, así como presentar por primera vez la implementación en hardware de dichos modelos. Los tres experimentos se realizaron para comparar el rendimiento entre las técnicas convencionales basadas en el algoritmo CORDIC y los modelos propuestos basados en la trigonometría racional.

De acuerdo con los experimentos, se ha observado que la mejora en términos de recursos computacionales es significativa, sin embargo, ese ahorro va disminuyendo conforme el número de GDL aumenta. Al tratar de generalizar nuestra metodología observamos que el número de EL se duplica para mecanismos de cuatro y cinco GDL, sin embargo, se planteó una implementación alternativa basada en el método MFF la cual no presenta ninguna ventaja en el ahorro de EL. Comparamos nuestros modelos con el método MFF, donde también se obtuvo una ventaja, debido al pre-procesamiento y post-procesamiento que exige dicho método. El ahorro de EL para el método MFF solo tendría relevancia cuando el mecanismo posea más de 10 GDL.

Nuestras implementaciones en hardware están optimizadas para mecanismos de cadena cinemática abierta de hasta tres GDL, las cuales se pueden utilizar en la mayoría de robots industriales de hasta seis GDL. El ahorro que se obtuvo de EL (CORDIC paralelo) así como la disminución de la latencia (CORDIC iterativo) se traducen en un menor consumo de potencia del circuito sintetizado. Adicionalmente un ahorro de casi 80% (mecanismo de un GDL) significa que con el mismo número de EL que se calcula la cinemática de un solo motor, ahora se podría calcular la cinemática de hasta cinco motores de forma simultánea con el mismo FPGA, con lo cual no sólo se podría reducir el consumo de potencia, sino que también se podría extender la capacidad de cómputo dedicada al cálculo de la cinemática hasta en cinco veces. Actualmente estamos trabajando en la obtención de modelos matemáticos

que permitan describir la orientación de forma racional, por lo que al combinar ambos trabajos se obtendría el modelado completo de un robot industrial de seis GDL.

AGRADECIMIENTOS

E. Zamora y H. Sossa desean agradecer el apoyo proporcionado por CIC-IPN para llevar a cabo esta investigación. Este trabajo fue apoyado económicamente por SIP-IPN (números de concesión 20180180, 20190166, 20180730 y 20190007), y el número de concesión CONACYT 65 (Frontiers of Science). R. Peralta reconoce a CONACYT por la beca otorgada para continuar sus estudios de doctorado.

REFERENCIAS

- [1] J. C. John, "Introduction to robotics: mechanics and control" *Pearson Education*, 2009.
- [2] P. P. Richard, "Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators," *MIT press*, 1981.
- [3] B. Luigi and M. Claudio, "Trajectory planning for automatic machines and robots," *Springer Science and Business Media*, 2008.
- [4] B. C. Eduardo, "Geometric computing: for wavelet transforms, robot vision, learning, control and action," *Springer Publishing Company Incorporated*, 2010.
- [5] H. Dietmar, "Foundations of geometric algebra computing," *AIP Conference Proceedings*. vol. 1479, no. 1, AIP, 2012.
- [6] P. Christian, et al, "Geometric algebra with applications in engineering," *Springer*, vol. no.4, Berlin, 2009.
- [7] V. Koulaei, S. Mojtaba and M. Rahimi. "Direct Kinematics Solution of 3-RCC Parallel Robot using a Semi-Analytical Homotopy Method," *International Journal of Advanced Design and Manufacturing Technology*, vol. 12, no. 1, 2019.
- [8] O. Altuzarra, D. Caballero, F. Campa and C. Pinto, "Position analysis in planar parallel continuum mechanisms," *Mechanism and Machine Theory*, vol. 132, pp. 13-29, 2019.
- [9] S. Kucuk and Z. Bingul, "Inverse kinematics solutions for industrial robot manipulators with offset wrists," *Applied Mathematical Modelling*, vol. 38, no 7-8, pp. 1983-1999, 2014.
- [10] W. J. Norman, "Divine Proportions: Rational trigonometry to universal geometry," *Wild Egg*, 2005.
- [11] P. A. Joao, "Rational Trigonometry Applied to Robotics," *Diss. Master Thesis*, Instituto Superior Técnico, Lisbon, Portugal, 2007.
- [12] P. A. Joao, "Rational Trigonometry Applied to Robotics: Robot Kinematic Modelling using Rational Trigonometry" 2007.
- [13] L. Junwei, F. Jiandong, L. Bajin, and Z. Yudong, "Study of CORDIC algorithm based on FPGA," *Chinese Control and Decision Conference (CCDC)*, pp. 4338-4343, May 2016.
- [14] W. Xiaoyuan, "Design and Implementation of CORDIC Algorithm Based on FPGA," *International Conference on Robots and Intelligent System (ICRIS)*, pp. 70-71, May 2018.
- [15] A. Supriya, K. M. Pramod, and K. Kavita, "Concept, design, and implementation of reconfigurable CORDIC," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no.4, pp. 1588-1592, 2016
- [16] M. Chinnathambi, N. Bharanidharan, S. Rajaram, "FPGA implementation of fast and area efficient CORDIC algorithm," *International Conference on Communication and Network Technologies*, pp. 228-232, December 2014.
- [17] W. Dong, M. Jean-Michel, B. Nicolas, D. E. Milos, "(M, p, k)-Friendly Points: A Table-Based Method to Evaluate Trigonometric Function," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 61, no. 9, pp. 711-715, 2014.
- [18] B. Ramesh, N. Parvathi, "Low power design techniques applied to pipelined parallel and iterative CORDIC design," *3rd International Conference on Electronics Computer Technology* vol. 5, pp. 336-340, April 2011.
- [19] W. J. Norman, "A rational approach to trigonometry," *Math Horizons*, vol. 15, no 2, pp. 16-20, 2007.
- [20] M. P. Rogelio, "Modelado de Robots con trigonometría Racional" *Diss. Doctoral Thesis*, 2019.
- [21] W. C. Chen, C. S. Chen, F. C. Lee, and Y. S. Kung, "Digital hardware implementation of the forward/inverse kinematics for a SCARA robot manipulator," *IEEE International Conference on Applied System Invention (ICASI)*, pp. 54-57, 2018.
- [22] K. F. Riley, P. H. Michael and J. B. Stephen, "Mathematical methods for physics and engineering: a comprehensive guide," *Cambridge university press*, 2006.
- [23] A. Supriya, K. M. Pramod, K. Kavita, "Concept, design and implementation of reconfigurable CORDIC," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, no. 4, pp. 1588-1592, 2016.
- [24] R. Bhakthavatchalu, M.S. Siniith, P. Nair, K. Jismi, "A Comparison of Pipelined Parallel and Iterative CORDIC Design on FPGA," *IEEE 5th International Conference on Industrial and Information Systems, ICIIIS*, pp. 239-243, 2010.
- [25] R. Bhakthavatchalu, P. Nair, "Low power design techniques applied to pipelined parallel and iterative CORDIC design," *IEEE 3rd International Conference on Electronics Computer Technology*, vol. 5, pp. 336-340, April 2011.
- [26] K. M. Pramod, et al, "50 years of CORDIC: Algorithms, architectures, and applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 9, pp. 1893-1907, 2009.
- [27] S. Tole, Jidin, Z. J. Aiman, J. Auzani and R. N. I. Nik. "Simplified VHDL coding of modified non-restoring square root calculator," *International Journal of Reconfigurable and Embedded Systems*, vol. 1, no. 1, pp. 37, 2012.
- [28] M. D. Ercegovic and L. Tomas, "Digital arithmetic," *Elsevier*, 2004.
- [29] J. M. Muller, "Elementary functions," *Birkhuser Boston*, 2006.
- [30] G. R. Rising, "Inside your calculator: from simple programs to significant insights," *John Wiley and Sons*, 2007.



Rogelio Martínez recibió el grado en Ingeniero en Electrónica y el grado de Maestro en ciencias en Microelectrónica por el IPN en México. Actualmente es candidato a Doctor en ingeniería por el CIC-IPN de México. Sus intereses de investigación incluyen el control no lineal, el modelado cinemático y la aritmética digital en FPGA.



Erik Zamora es profesor titular en el IPN de México. Recibió un Diploma en electrónica de UV (2004), una Maestría en Ingeniería Eléctrica de CINVESTAV-IPN (2007) y un Doctorado en control automático del CINVESTAV-IPN (2015). Desarrolló el primer sistema comercial mioeléctrico mexicano para controlar una prótesis en la compañía PRO/BIONICS y un sistema de navegación robótica para entornos desconocidos guiados por señales de emergencia en la Universidad de Bristol. Sus intereses actuales incluyen robots autónomos y aprendizaje automático. Ha publicado más de veintidós artículos técnicos en conferencias y revistas internacionales y ha dirigido más de treinta tesis.



Humberto Sossa nació en Guadalajara, Jalisco, México en 1956. Recibió el título de Ingeniero en Electrónica por la Universidad de Guadalajara (1981), el grado de maestro en ciencias en Ingeniería Eléctrica en CINVESTAV-IPN (1987) y un Doctorado en Informática del Instituto Politécnico Nacional de Grenoble, Francia (1992). Es profesor de tiempo completo en el Centro de Investigación en Computación del Instituto Politécnico Nacional de México. Sus principales áreas de investigación son las de patrón. Reconocimiento, redes neuronales, análisis de imágenes y control de robots mediante el análisis de imágenes.

TEOREMAS DE EQUIVALENCIA

Los siguientes teoremas de equivalencia están desarrollados con base en las 5 leyes principales de la trigonometría racional, así como en algunas propiedades trigonométricas clásicas. No se desarrollaran los procedimientos que consideremos elementales ó resultados que consideremos obvios de interpretar.

CINEMÁTICA DIRECTA 1 GDL

Las coordenadas q_x y q_y para el mecanismo de la Figura 2(b) están dadas en forma racional por las siguientes expresiones

$$q_x = Q_1(1 - s_1) \tag{61}$$

$$q_y = Q_1 s_1 \tag{62}$$

por definición $s_1 = \frac{q_y}{Q_1}$, por lo que aplicando raíz cuadrada a la ecuación (62) se obtiene

$$\sqrt{q_y} = \sqrt{Q_1 s_1} = \sqrt{Q_1} \sqrt{\frac{q_y}{Q_1}}$$

ya que por definición el quadrance es equivalente a la norma euclidiana elevada al cuadrado se tiene que $L_1 = \sqrt{Q_1}$ y $y = \sqrt{q_y}$, por lo tanto

$$y = \sqrt{q_y} = L_1 \frac{y}{L_1} \tag{63}$$

la función trigonométrica que relaciona a y y L_1 de acuerdo con la Figura 2(a) es el $\sin \theta_1$, por lo que la ecuación anterior se puede reescribir como

$$y = \sqrt{q_y} = L_1 \sin \theta_1 \tag{64}$$

De manera similar aplicando raíz cuadrada a la ecuación (61) se obtiene la equivalencia para la coordenada x

$$\sqrt{q_x} = \sqrt{Q_1(1 - s_1)} = \sqrt{Q_1} \sqrt{1 - s_1} \tag{65}$$

$$x = L_1 \cos \theta_1 \tag{66}$$

La equivalencia trigonométrica $1 - s_1 = \cos^2 \theta_1$ se debe a que $(1 - s_1)$ es el ángulo complementario de s_1 .

CINEMÁTICA INVERSA 1 GDL

El parámetro articular en forma racional está dado por

$$s_1 = \frac{q_y}{q_x + q_y} \tag{67}$$

utilizando el teorema de Pitágoras racional se tiene que $Q_1 = q_x + q_y$, por lo que la expresión anterior se convierte en

$$s_1 = \frac{q_y}{Q_1} \tag{68}$$

ya que por definición el quadrance es equivalente a la norma euclidiana elevada al cuadrado se tiene que

$$s_1 = \frac{y^2}{L_1^2} = \left(\frac{y}{L_1} \right)^2 \tag{69}$$

por lo que de la expresión anterior se concluye que

$$s_1 = \sin^2 \theta$$

CINEMÁTICA DIRECTA 2 GDL

Las coordenadas q_x y q_y para el mecanismo de la Figura 4 están dadas en forma racional por las siguientes expresiones

$$q_x = q_3(1 - s_a) \tag{70}$$

$$q_y = q_3 s_a \tag{71}$$

Para obtener la equivalencia de la coordenada q_y es necesario expandir algebraicamente la ecuación (71) como se muestra a continuación

$$q_y = q_3 s_a \tag{72}$$

$$= q_3 \left(\sqrt{s_1(1 - \frac{s_2 Q_2}{q_3})} - \sqrt{\frac{s_2 Q_2}{q_3}(1 - s_1)} \right)^2 \tag{73}$$

$$= q_3 \left(s_1(1 - \frac{s_2 Q_2}{q_3}) - 2\sqrt{s_1(1 - s_1)(1 - \frac{s_2 Q_2}{q_3})(\frac{s_2 Q_2}{q_3})} + \frac{s_2 Q_2}{q_3}(1 - s_1) \right) \tag{74}$$

$$= s_1 q_3 + s_2 Q_2 - 2s_1 s_2 Q_2 - 2q_3 \sqrt{s_1(1 - s_1)(q_3 - s_2 Q_2)s_2 Q_2} \tag{75}$$

sustituyendo el valor de q_3 de acuerdo con la ley cruz definida como $q_3 = Q_1 + Q_2 - 2\sqrt{Q_1 Q_2(1 - s_2)}$ se obtiene

$$= s_2 Q_2 - 2s_1 s_2 Q_2 + s_1 q_3 - 2\sqrt{s_1(1 - s_1)s_2 Q_2} \sqrt{q_3 - s_2 Q_2} \tag{76}$$

$$= s_2 Q_2 - 2s_1 s_2 Q_2 + s_1 \left(Q_1 + Q_2 - 2\sqrt{s_1 s_2 Q_2(1 - s_2)} \right) \tag{77}$$

$$- 2\sqrt{s_1(1 - s_1)s_2 Q_2} \sqrt{\left(Q_1 + Q_2 - 2\sqrt{Q_1 Q_2(1 - s_2)} \right) - s_2 Q_2}$$

$$= s_1 Q_1 + s_1 Q_2 + s_2 Q_2 - 2s_1 s_2 Q_2 - 2s_1 \sqrt{s_1 s_2 Q_2(1 - s_2)} \tag{78}$$

$$- 2\sqrt{s_1(1 - s_1)s_2 Q_2} \sqrt{\left(\sqrt{Q_1} - \sqrt{Q_2(1 - s_2)} \right)^2}$$

$$= s_1 Q_1 + s_1 Q_2 + s_2 Q_2 - 2s_1 s_2 Q_2 - 2s_1 \sqrt{s_1 s_2 Q_2(1 - s_2)} \tag{79}$$

$$- 2\sqrt{s_1 s_2(1 - s_1)Q_1 Q_2} + 2Q_2 \sqrt{s_1 s_2(1 - s_1)(1 - s_2)}$$

A partir de aquí se simplifican las ecuaciones anteriores para llevarlas a su equivalente en funciones trigonométricas

$$= s_1 Q_1 + 2s_1 \sqrt{Q_1 Q_2(1 - s_2)} + 2\sqrt{Q_1 Q_2 s_1 s_2(1 - s_1)} \tag{80}$$

$$+ Q_2 s_1(1 - s_2) + 2Q_2 \sqrt{s_1 s_2(1 - s_1)(1 - s_2)} + Q_2 s_2(1 - s_1)$$

sustituyendo los valores de s_1 y s_2 por sus equivalentes trigonométricos en la ecuación anterior se obtiene lo siguiente

$$= \sin \theta_1 Q_1 + 2\sqrt{Q_1 Q_2} \sin \theta_1 (\sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2) \tag{81}$$

$$+ Q_2 (\sin \theta_1 \cos \theta_2 + \sin \theta_2 \cos \theta_1)^2$$

reemplazando los quadrance por las distancias euclidianas

$$y^2 = L_1^2 \sin^2 \theta_1 + 2L_1 L_2 \sin \theta_1 \sin(\theta_1 + \theta_2) + L_2^2 \sin^2(\theta_1 + \theta_2) \tag{82}$$

$$y^2 = (L_1^2 \sin \theta_1 + L_2 \sin^2(\theta_1 + \theta_2))^2 \tag{83}$$

finalmente aplicando raíz cuadrada a la ecuación anterior se obtiene

$$y = L_1 \sin \theta_1 + L_2 \sin(\theta_1 + \theta_2) \tag{84}$$

De forma similar al procedimiento anterior se desarrolla la expresión para la coordenada q_x

$$q_x = q_3(1 - s_a) \tag{85}$$

$$= q_3 - q_3 \left(\sqrt{s_1(1 - \frac{s_2 Q_2}{q_3})} - \sqrt{\frac{s_2 Q_2}{q_3}(1 - s_1)} \right)^2 \tag{86}$$

$$= q_3 - q_3 \left(s_1(1 - \frac{s_2 Q_2}{q_3}) - 2\sqrt{s_1(1 - \frac{s_2 Q_2}{q_3})(\frac{s_2 Q_2}{q_3}(1 - s_1))} + \frac{s_2 Q_2}{q_3}(1 - s_1) \right) \tag{87}$$

$$+ \frac{s_2 Q_2}{q_3}(1 - s_1)$$

$$= (1 - s_1)q_3 + (2s_1 - 1)s_2 Q_2 - 2\sqrt{s_1(1 - s_1)(q_3 - s_2 Q_2)s_2 Q_2} \tag{88}$$

sustituyendo el valor de q_3 de acuerdo con la ley cruz definida como $q_3 = Q_1 + Q_2 - 2\sqrt{Q_1 Q_2(1-s_2)}$ se obtiene

$$q_x = (1-s_1) \left(Q_1 + Q_2 - 2\sqrt{Q_1 Q_2(1-s_2)} \right) + (2s_1-1)s_2 Q_2 \quad (89)$$

$$- 2\sqrt{s_1(1-s_1)s_2 Q_2} \sqrt{Q_1 + Q_2 - 2\sqrt{Q_1 Q_2(1-s_2)} - s_2 Q_2} \\ = +2\sqrt{s_1(1-s_1)s_2 Q_2} (\sqrt{Q_1} - \sqrt{(1-s_2)Q_2}) \quad (90)$$

$$(1-s_1)(Q_1 + Q_2) + s_2 Q_2(2s_1-1) - 2(1-s_1)\sqrt{Q_1 Q_2(1-s_2)} \\ = Q_1(1-s_1) + Q_2 s_1 s_2 + Q_2(1-s_1)(1-s_2) \quad (91)$$

$$- 2(1-s_1)\sqrt{Q_1 Q_2(1-s_2)} + 2\sqrt{Q_1 Q_2 s_1 s_2(1-s_1)} \\ - 2Q_2 \sqrt{s_1 s_2(1-s_1)(1-s_2)}$$

$$= Q_1(1-s_1) + 2(1-s_1)\sqrt{Q_1 Q_2(1-s_2)} - 2\sqrt{Q_1 Q_2 s_1 s_2(1-s_1)} \\ + Q_2(1-s_1)(1-s_2) - 2Q_2 \sqrt{s_1 s_2(1-s_1)(1-s_2)} + Q_2 s_1 s_2 \quad (92)$$

sustituyendo los valores de s_1 y s_2 por sus equivalentes trigonométricos en la ecuación anterior se obtiene lo siguiente

$$q_x = Q_1 \cos^2 \theta_1 + 2\sqrt{Q_1 Q_2} \cos^2 \theta_1 \cos^2 \theta_2 + Q_2 \cos^2 \theta_1 \cos^2 \theta_2 \quad (93)$$

$$- 2\sqrt{Q_1 Q_2} \cos \theta_1 \sin \theta_1 \sin \theta_2 + Q_2 \sin^2 \theta_1 \sin^2 \theta_2 \\ - 2Q_2 \cos \theta_1 \cos \theta_2 \sin \theta_1 \sin \theta_2$$

$$x^2 = (L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2))^2 \quad (94)$$

reemplazando los quadrance por las distancias euclidianas

$$x^2 = L_1^2 \cos^2 \theta_1 + 2L_1 L_2 \cos^2 \theta_1 \cos^2 \theta_2 + L_2^2 \cos^2 \theta_1 \cos^2 \theta_2 \quad (95)$$

$$- 2L_1 L_2 \cos \theta_1 \sin \theta_1 \sin \theta_2 + L_2^2 \sin^2 \theta_1 \sin^2 \theta_2 \\ - 2L_2^2 \cos \theta_1 \cos \theta_2 \sin \theta_1 \sin \theta_2$$

$$x^2 = (L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2))^2 \quad (96)$$

aplicando raíz cuadrada a la ecuación anterior se obtiene

$$x = L_1 \cos \theta_1 + L_2 \cos(\theta_1 + \theta_2)$$

CINEMÁTICA INVERSA 2 GDL

Los parámetros articulares para el mecanismo de dos GDL mostrado en la Figura 4 se calculan con las ecuaciones

$$s_1 = s'_a \oplus s_a \quad (97)$$

$$s_2 = 1 - \frac{(Q_1 + Q_2 - q_3)^2}{4Q_1 Q_2} \quad (98)$$

La equivalencia del segundo parámetro articular s_2 se obtiene despejando el término fraccionario y aplicando raíz cuadrada, como se muestra a continuación

$$\sqrt{1-s_2} = \frac{Q_1 + Q_2 - q_3}{2\sqrt{Q_1 Q_2}} \quad (99)$$

teniendo en cuenta la equivalencia $1-s_2 = \cos^2 \theta_2$ y reemplazando el valor de los quadrance por sus equivalentes euclidianos se obtiene

$$\cos \theta_2 = \frac{L_1^2 + L_2^2 - R^2}{2L_1 L_2} \quad (100)$$

la expresión anterior es la misma que se utiliza habitualmente para calcular la cinemática inversa del mecanismo mostrado en la Figura 3.

La equivalencia del parámetro s_1 se obtiene reemplazando las variables s'_a y s_a en la ecuación (97), donde s'_a se obtiene

utilizando la ley de spread y s_a se obtiene utilizando la definición de spread.

$$s_1 = \frac{s_2 Q_2}{q_3} \oplus \frac{q_y}{q_3} \quad (101)$$

en la ecuación anterior el símbolo \oplus corresponde a la operación suma de spreads, el cual será explicado al final de esta demostración, por tanto, analizaremos los términos de la ecuación anterior por separado.

La equivalencia para el primer término $s'_a = \frac{s_2 Q_2}{q_3}$ se obtiene aplicándole raíz cuadrada, como se muestra a continuación

$$\sqrt{s'_a} = \frac{\sqrt{s_2} \sqrt{Q_2}}{\sqrt{q_3}} \quad (102)$$

dato que $s'_a = \sin^2 \beta$ la equivalencia trigonométrica de acuerdo con la Figura 3 sería la siguiente

$$\sin \beta = \frac{\sin \theta_2 L_2}{R} \quad (103)$$

a la relación anterior se le conoce como la ley de senos, por lo que la ley de spread es equivalente a la ley de senos. Para obtener la equivalencia del segundo término de la ecuación (99) se le aplica raíz cuadrada

$$\sqrt{s_a} = \frac{\sqrt{q_y}}{\sqrt{q_3}} \Rightarrow \sin \alpha = \frac{y}{R} \quad (104)$$

es evidente que la ecuación (101) se puede reescribir como

$$\sin^2 \theta_1 = \sin^2 \alpha \oplus \sin^2 \beta \quad (105)$$

la ecuación anterior es equivalente a la función trigonométrica

$$\sin \theta_1 = \sin \alpha + \sin \beta \quad (106)$$

dicha equivalencia se obtiene aplicando la definición del símbolo \oplus , la cual se demuestra a continuación.

SUMA DE SPREADS

Adicionalmente al procedimiento anterior es necesario definir los símbolos \oplus y \ominus como la suma y resta de spreads, los cuales no corresponden a la suma y resta que habitualmente conocemos. Analizando la Figura 13 se puede deducir como calcular la suma de spreads. Las expresiones matemáticas que

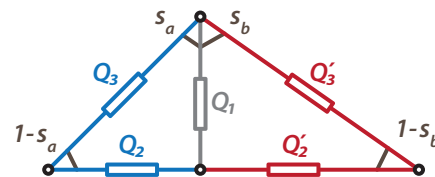


Fig. 13. Diagrama para el análisis de la suma de spreads

permiten desarrollar sumas y restas de spreads están dadas por

$$s_a \oplus s_b = \left(\sqrt{s_a(1-s_b)} + \sqrt{s_b(1-s_a)} \right)^2 \quad (107)$$

$$s_a \ominus s_b = \left(\sqrt{s_a(1-s_b)} - \sqrt{s_b(1-s_a)} \right)^2 \quad (108)$$

de acuerdo con la Figura 13, usando la ley de spread se obtiene

$$\frac{s_a \oplus s_b}{Q_2 \boxplus Q'_2} = \frac{1 - s_b}{Q_3} = \frac{1 - s_a}{Q'_3} \quad (109)$$

de la ecuación (109) se deducen las siguientes dos ecuaciones

$$s_a \oplus s_b = \frac{(1 - s_b)(Q_2 \boxplus Q'_2)}{Q_3} \quad (110)$$

$$s_a \oplus s_b = \frac{(1 - s_a)(Q_2 \boxplus Q'_2)}{Q'_3} \quad (111)$$

dado que ambas expresiones son equivalentes se afirma que

$$\frac{(1 - s_b)(Q_2 \boxplus Q'_2)}{Q_3} = \frac{(1 - s_a)(Q_2 \boxplus Q'_2)}{Q'_3} \quad (112)$$

usando la definición de spread y reemplazando Q_3 y Q'_3 en la ecuación anterior se obtiene la siguiente expresión

$$\frac{(1 - s_b)(1 - s_a)(Q_2 \boxplus Q'_2)}{Q_1} = \frac{(1 - s_a)(1 - s_b)(Q_2 \boxplus Q'_2)}{Q_1} \quad (113)$$

por consiguiente, simplificando y igualando la ecuación anterior con la ecuación (111) se obtiene

$$s_a \oplus s_b = Q_2 \boxplus Q'_2 = \left(\sqrt{Q_2} + \sqrt{Q'_2} \right)^2 \quad (114)$$

los valores de Q_2 y Q'_2 se calculan como

$$Q_2 = \left[\frac{Q'_3}{1 - s_a} \right] (1 - s_b) s_a \quad (115)$$

$$Q'_2 = \left[\frac{Q_3}{1 - s_b} \right] (1 - s_a) s_b \quad (116)$$

en las ecuaciones (115) y (116) ambos términos entre paréntesis cuadrados son equivalentes. La ecuación (117) se obtiene reemplazando estos términos en la ecuación (111) y realizando algunas operaciones algebraicas

$$s_a \oplus s_b = \left[\frac{Q'_3}{1 - s_a} \right] \left(\sqrt{s_a(1 - s_b)} + \sqrt{s_b(1 - s_a)} \right)^2 \quad (117)$$

la expresión entre paréntesis cuadrado puede ser reemplazada por la ecuación (109)

$$s_a \oplus s_b = \left[\frac{Q_2 \boxplus Q'_2}{s_a \oplus s_b} \right] \left(\sqrt{s_a(1 - s_b)} + \sqrt{s_b(1 - s_a)} \right)^2 \quad (118)$$

finalmente, moviendo el denominador al lado derecho y tomando la raíz cuadrada a ambos lados de la ecuación, obtenemos el resultado. Para la resta de spreads, el procedimiento es similar pero en lugar de usar \boxplus se usará \boxminus .

$$s_a \oplus s_b = \left(\sqrt{s_a(1 - s_b)} + \sqrt{s_b(1 - s_a)} \right)^2 \quad (119)$$

CINEMÁTICA PARA MECANISMOS DE 3 GDL

Los teoremas de equivalencia para los mecanismos de 3 GDL siguen el mismo procedimiento y utilizan las mismas fórmulas que los cuatro teoremas enunciados anteriormente, por dicha razón hemos decidido no ponerlos. Sin embargo para que el lector no tenga que consultar todas las referencias bibliográficas, a continuación se enuncian las cinco leyes principales de la trigonometría racional.

Teorema de Pitágoras: El teorema de Pitágoras de forma racional se define de acuerdo con la siguiente ecuación y solo se puede aplicar cuando $A_1A_3 \perp A_3A_2$

$$Q_3 = Q_1 + Q_2 \quad (120)$$

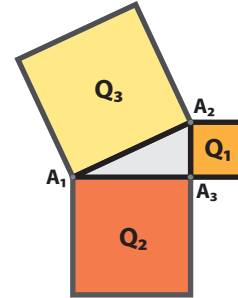


Fig. 14. Diagrama para el análisis de la suma de spreads

Fórmula de triple quad: A diferencia del teorema de Pitágoras, para aplicar la fórmula del triple quad, se debe verificar que A_1, A_2 y A_3 sean colineales.

$$(Q_1 + Q_2 - Q_3)^2 = 4Q_1Q_2 \quad (121)$$

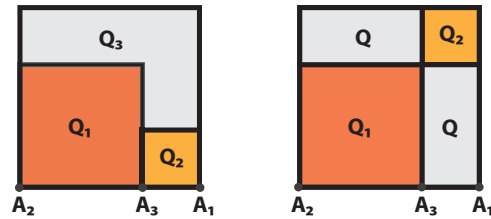


Fig. 15. Diagrama para el análisis de la suma de spreads

Ley de spread: La ley de spread que se muestra en la ecuación (122) y se corresponde con la ley de senos en la trigonometría clásica. Esta ley da una relación entre los spreads y los cuadrantes de un triángulo.

$$\frac{s_1}{Q_1} = \frac{s_2}{Q_2} = \frac{s_3}{Q_3} \quad (122)$$

Ley cruz: La ley cruz se corresponde con la ley de cosenos en la trigonometría clásica. La ley cruz se define de acuerdo con la siguiente ecuación.

$$(Q_1 + Q_2 - Q_3)^2 = 4Q_1Q_2(1 - s_3) \quad (123)$$

Fórmula de triple spread: La fórmula de triple spread que se muestra en la ecuación (124) es similar a la ley cruz. En este caso, los valores para los cuadrantes se intercambian por sus spreads correspondientes.

$$(s_1 + s_2 - s_3)^2 = 4s_1s_2(1 - s_3) \quad (124)$$