



UNIVERSIDAD TECNOLÓGICA DE BRNO

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

FACULTAD DE INGENIERÍA MECÁNICA

FAKULTA STROJNÍHO INŽENÝRSTVÍ

INSTITUTO DE MATEMÁTICAS

ÚSTAV MATEMATIK

APLICACIONES DE CUATERNIONES EN ROBOT CINEMÁTICA

APLIKACE QUATERNION® V KINEMATICE ROBOTU

TESIS DE MAESTRÍA

DIPLOMOVÁ PRÁCE

AUTOR

AUTOR PRÁCE

doctora diana

SUPERVISOR

VEDOUCÍ PRÁCE

doc. En g. Radomil Matoušek, Ph.D.

BRNO 2019



Especificación Trabajo de Fin de Máster

Departamento: instituto de matematicas
Alumno: doctora diana
Programa de estudio: Ciencias Aplicadas en Ingeniería
Campo de estudio: Ingeniería Matemática
Supervisor: doc. En g. Radomil Matoušek, Ph.D.
Año académico: 2018/19

De conformidad con la Ley núm. 111/1998 relativo a las universidades y las normas de estudio y examen BUT, usted el director del instituto les ha asignado el siguiente tema de tesis de maestría:

Aplicaciones de los cuaterniones en cinemática de robots

Característica concisa de la tarea:

Esta tesis trata sobre la utilidad de la representación cuaterniana de la cinemática para la robótica. aplicaciones Tanto la cinemática directa como la inversa para un manipulador robótico se derivan usando cuaterniones.

Objetivos Trabajo Fin de Máster:

La revisión de robots 6DOF y modelos cinemáticos directos e inversos para el diseño de control.
Estado del arte en control de robots usando álgebra de cuaterniones.
Simulación de robot (Universal Robot) y aplicación práctica de programación.

Bibliografía recomendada:

SICILIANO, Bruno a KHATIB, Oussama, ed. Springer manual de robótica. 2ª edición. Berlina: Springer, [2016]. ISBN 978-3-319-32550-7.

La fecha límite para la presentación de la Tesis de Maestría está dada por el Calendario del año académico 2018/19

en brno,

LS

profe. RNDr. Josef Šlapal, CSc.
Director del Instituto

doc. En g. Jaroslav Katolický, Ph.D.
decano de la FME

Abstracto

Esta tesis trata sobre la utilidad de la aplicación de cuaterniones en la representación de la cinemática de robots. Comienza mostrando la relación de los cuaterniones con los números complejos más conocidos y cómo puede representar rotaciones en tres dimensiones. Luego, se introducen los cuaterniones duales para representar tanto la rotación como la traslación tridimensionales. Luego se usará para derivar la cinemática directa e inversa, en particular, para el Universal Robot UR3, que es un brazo robótico de 6 grados de libertad. Por último, se demostrará una aplicación real de cuaterniones duales en la programación de robots.

Palabras clave

cuaterniones, cuaterniones duales, robótica, robot universal, UR3, cinemática directa, cinemática inversa, rotaciones tridimensionales

DOCTOR, DB Aplicaciones de los Cuaterniones en Cinemática de Robots. Brno: Universidad Tecnológica de Brno, Facultad de Ingeniería Mecánica, 2019. 105p. Directora de tesis de diploma prof. doc. En g. Radomil Matoušek, Ph.D.

Declaro que he escrito esta tesis de diploma Aplicaciones de cuaterniones en cinemática de robots por mi cuenta bajo la dirección de mi supervisor, el prof. doc. En g. Radomil Matoušek, Ph.D., y la asistencia del Ing. Roman Parák, y utilizando las fuentes citadas en la bibliografía.

31 de mayo de 2019

Diana B. Doctora

Quisiera dedicar esta tesis, primero, a mi padre, Mateo, quien me crió como la persona que soy hoy, llena de sueños y esperanzas, y a las personas detrás del programa InterMaths, quienes me dieron múltiples oportunidades para cumplir estos sueños.

Diana B. Doctora

Contenido

1. Introducción	13
2 Números Complejos 2.1	15
Historia de los Números Complejos.	15
2.2 El sistema de números complejos.	dieciséis
2.2.1 Definición.	dieciséis
2.2.2 Multiplicación de números complejos.	dieciséis
2.3 Representación gráfica de $z \in \mathbb{C}$	17
2.4 Propiedades de los Números Complejos.	18
2.4.1 Norma y Argumento de $z \in \mathbb{C}$	18
2.4.2 Complejo Conjugado de $z \in \mathbb{C}$	18
2.4.3 Reglas algebraicas para números complejos.	19
2.4.4 Adición de Números Complejos.	20
2.4.5 Multiplicación de Números Complejos.	20
2.5 Rotaciones en el Plano Complejo.	22
3 Cuaterniones 3.1	23
Historia de los Cuaterniones.	23
3.2 El Sistema Numérico Quaternion.	26
3.2.1 Definición.	26
3.2.2 Representaciones de cuaterniones.	26
3.2.3 Leyes de multiplicación de cuaterniones.	27
3.3 Propiedades de los cuaterniones.	28
3.3.1 Conjugado de cuaterniones.	28
3.3.2 Norma de un Cuaternión.	28
3.3.3 Unidad Cuaternión.	28
3.3.4 Inverso de un Cuaternión.	28
3.3.5 Suma y Resta de Cuaterniones.	29
3.3.6 Multiplicación y División de Cuaterniones.	29
3.3.7 Propiedades resumidas de los cuaterniones.	30
3.4 Rotaciones 3D usando cuaterniones.	31

3.4.1 Representaciones de rotación 3D.	31
3.4.2 Conversiones de cuaterniones.	36
3.5 Cuaterniones duales.	38
3.5.1 Definición.	38
3.5.2 Inverso de un Cuaternión Dual.	38
3.5.3 Multiplicación de un Cuaternión Dual. . .	39
3.5.4 Transformación por un Cuaternión Dual. .	39
4 Aplicación de cuaterniones a Universal Robot UR3	41
4.1 Robot Universal - UR3	41
4.2 Cinemática directa por cuaterniones duales.	42
4.2.1 Representaciones de la Transformación de Articulaciones.	42
4.2.2 Cálculo de cinemática directa UR3 . .	44
4.3 Cinemática inversa por cuaterniones duales.	45
4.3.1 Representaciones de transformaciones de articulaciones.	45
4.3.2 Productos de transformación de juntas.	46
4.3.3 Transformaciones de armazones de la base a las muñecas	47
4.3.4 Transformación de marcos de hombro a muñeca.	53
4.3.5 Soluciones de ángulos de articulación.	56
4.4 Aplicación práctica de programación.	57
4.4.1 Parámetros de Denavit-Hartenberg.	57
4.4.2 Aplicación de programación Python.	58
4.5 Verificación de Resultados.	71
5 Conclusiones y Recomendaciones	75
5.1 Conclusión.	75
5.2 Recomendación.	77
Apéndices	79
Un robot universal UR3	
Especificaciones técnicas	81
Productos de transformación B para Base toWrist 3	85
Productos de transformación C para hombro a muñeca 1	97
D Lista de tablas y figuras	101

Capítulo 1

Introducción

El concepto de robótica ha existido incluso durante la antigüedad. Sin embargo, el término "robot" se concibió recién en el siglo XX y se originó en el país de la República Checa. Fue introducido por primera vez por los checos

el dramaturgo Karel Capek en su obra "Rossum's Universal Robots (RUR)" que se estrenó en la ciudad de Praga en la década de 1920. Sin embargo, fue el hermano de Karel, Josef Capek, quien realmente inventó el término "robot" y lo derivó de la palabra eslava "robota" que significa trabajo subordinado o forzado [2]. La obra se volvió rápidamente influyente y

luego se agregó el término "robot" al idioma inglés y desde entonces también se ha utilizado en ficciones y en investigaciones científicas. En la era moderna, nos referimos a un robot como una máquina hecha por el hombre que puede realizar trabajos e imitar acciones realizadas por humanos. Hemos visto varias aplicaciones de robots que son más comunes en la industria manufacturera. La aplicación más avanzada de ellos serían las exploraciones marítimas y espaciales.

Modelar un robot es donde entra en juego la cinemática del robot. Formular la cinemática correcta de un robot es importante para analizar su comportamiento y manipularlo. Los robots están compuestos de articulaciones y enlaces. Los dos tipos más comunes de articulaciones son las articulaciones angulares y prismáticas. Estas articulaciones realizan rotaciones y traslaciones, respectivamente. Podemos comparar las articulaciones giratorias con las articulaciones humanas, como las articulaciones del hombro, el codo, las muñecas y las rodillas. Un robot también se define por sus grados de libertad. Esto corresponde a todos los posibles movimientos de sus articulaciones. Un robot plano se describe con tres grados de libertad. Esto se debe a que solo puede moverse en un plano.

Estos son izquierda o derecha y hacia adelante o hacia atrás. El otro grado de libertad es la rotación que se puede realizar en un plano. Si un robot también puede hacer un movimiento hacia arriba y hacia abajo, que es un eje de movimiento adicional, ahora tiene cuatro grados de libertad. Si un robot también puede cambiar su pose (posición

Capítulo 1 Introducción

y orientación) mediante la rotación sobre otro eje en el plano, entonces también se suma a sus grados de libertad. Estas rotaciones se conocen más comúnmente como cabeceo, guiñada y balanceo. Si un robot puede hacer todas estas traslaciones y rotaciones, entonces tiene el máximo de seis grados de libertad.¹ La cinemática

de robots es el estudio de todos los movimientos posibles de un robot sin considerar las fuerzas que causan estos movimientos, como se establece en [21]. y [12].

La cinemática se puede dividir en cinemática directa e inversa. Forward Kinematics

determina la pose del efector final dados los ángulos de las articulaciones. Esto se puede derivar fácilmente y la solución siempre es única. Sin embargo, su complemento, la cinemática inversa, es generalmente mucho más difícil de derivar, ya que determina los ángulos de articulación necesarios para una determinada pose. Las ecuaciones son no lineales y pueden tener singularidades. Aparte de esto, también pueden existir múltiples soluciones. Nuestra aplicación implica la verificación de estas soluciones; sin embargo, determinar la solución óptima no está dentro del alcance del estudio.

La Cinemática del Robot se describe principalmente por la transformación que realiza cada una de las articulaciones del robot. Estas transformaciones se descomponen en rotaciones y traslaciones. Una articulación robótica solo puede realizar una traslación, como las articulaciones prismáticas, lo que significa que no afecta la orientación del efector final, solo su posición. También hay una articulación que puede realizar tanto como articulaciones giratorias. Estas articulaciones no solo cambian la orientación del efector final sino que también pueden cambiar su posición. Un ejemplo es si esta articulación está conectada a un enlace con el efector final. Este vínculo constituye la traducción entre este marco conjunto y el marco del efector final. Por lo tanto, la rotación de la articulación también afectará la longitud de la traslación con respecto al marco de coordenadas de la base, así como la orientación del efector final.

Para las rotaciones, hay varias formas de representarlas. Las formas más comunes son las matrices de rotación, los ángulos de Euler, la representación del eje-ángulo y los cuaterniones unitarios. Este estudio se centra en los cuaterniones unitarios, lo que nos brinda una representación elegante y compacta de la rotación. Mientras que una matriz de rotación de 3×3 necesita 9 elementos para representar una rotación, los cuaterniones solo necesitan 4. Sin embargo, esta representación todavía no se usa tanto como las transformaciones homogéneas en robótica. Esto quizás por el poco conocimiento que tenemos al respecto. Los cuaterniones son una extensión de los números complejos. Por lo tanto, para comprender la idea básica de los cuaterniones, primero debemos comprender el concepto de números complejos. El propósito de esta investigación es facilitar la comprensión de la relación entre los cuaterniones y los números complejos y mostrar cómo los cuaterniones, más específicamente los cuaterniones duales, pueden representar mejor las transformaciones en el espacio tridimensional.

¹Del curso edX: "Robótica: cinemática y fundamentos matemáticos" en [1]

Capítulo 2

Números complejos

Antes de comenzar el estudio de los cuaterniones, necesitaremos recordar algunas ideas básicas y algunas propiedades importantes de los números complejos que son muy beneficiosas para comprender el concepto de cuaterniones. Primero comenzaremos explicando la razón por la cual se formularon los números complejos y luego pasaremos a describir este sistema numérico. Luego, discutiremos las propiedades importantes de los números complejos definiéndolos en el sentido algebraico y también en el sentido geométrico.

1

2.1 Historia de los números complejos

El sistema de números reales $\mathbb{R}$ resultó de la búsqueda de un sistema que incluyera los números racionales y proporcionara soluciones a varias ecuaciones polinómicas. Sin embargo, hay algunas ecuaciones polinómicas como $x^2 + 2x + 2 = 0$ que no pueden ser satisfechas por ningún número real x . Usando la fórmula cuadrática $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$ expresiones formales para las dos soluciones para el sistema $ax^2 + bx + c = 0$; pero en los casos de las ecuaciones anteriores, esta fórmula involucra raíces cuadradas de números negativos que antes no estaban definidos. Para el siglo XVI, el matemático italiano Geronimo Cardano, comenzó a considerar este tipo de ecuaciones y enfatizó la necesidad tanto de números negativos como de "números complejos". Se dio cuenta de que si estas expresiones fueran a ser factorizadas por $\sqrt{-1}$ con la propiedad de que $\sqrt{-1} \cdot \sqrt{-1} = -1$, ciertamente satisficieran como soluciones a estas ecuaciones. Ahora usamos la designación ampliamente aceptada $i = \sqrt{-1}$ para estas expresiones tales que $i = -1$.

¹Todos los contenidos de este capítulo, incluidas las imágenes, están referenciados en el libro "Análisis complejo básico" de JE Marsden y MJ Hoffman [18]

2.2 El sistema de números complejos

Comenzamos definiendo un número complejo y explicando la regla de multiplicación de números complejos que es necesaria para comprender la multiplicación de cuaterniones.

2.2.1 Definición

El sistema de números complejos, denotado por $\mathbb{C}$, es el conjunto $\mathbb{R}^2$ el plano xy que consta de todos los pares ordenados (x, y) de números reales junto con las reglas habituales de suma de vectores,

$$(x_1, y_1) + (x_2, y_2) = (x_1 + x_2, y_1 + y_2) \quad (2.1)$$

y la multiplicación escalar por un número real a ,

$$a(x, y) = (ax, ay) \quad (2.2)$$

y con la operación añadida de multiplicación compleja definida por,

$$(x_1, y_1)(x_2, y_2) = (x_1x_2 - y_1y_2, x_1y_2 + y_1x_2) \quad (2.3)$$

Para la representación de números complejos, primero identificamos los números reales x como puntos en el eje real x e y como los puntos en el eje imaginario y donde el punto unidad $(0, 1)$ representa i . Ahora, podemos representar un número complejo como $(x, y) = x + iy$, que es la notación más estándar. Por lo tanto, un solo símbolo como $z = a + ib$ también se puede usar para indicar un número complejo donde $\text{Re} z = a$ se denomina parte real e $\text{Im} z = b$ se denomina parte imaginaria. Por tanto, un número real es un número complejo z donde $\text{Im} z = b = 0$.

Si tenemos el opuesto, $\text{Re} z = a = 0$, entonces tenemos un número imaginario puro. La notación $z \in \mathbb{C}$ implica que z pertenece al conjunto de los números complejos.

2.2.2 Multiplicación de números complejos

Para derivar la regla para la multiplicación de números complejos,

$$(a, b)(c, d) = (ac - bd, ad + bc) \quad (2.4)$$

usamos las notaciones estándar para los números complejos (a, b) y (c, d) , y luego aplicando la propiedad $i^2 = -1$, obtenemos:

$$\begin{aligned} (a + ib)(c + id) &= ac + iad + ibc + i^2bd \\ &= ac + i(ad + bc) + (-1)bd = ac - bd + i(ad + bc) \end{aligned} \quad (2.5)$$

2.3 Representación gráfica de z C

Como hemos dicho en la definición, un número complejo también se puede considerar como un vector bidimensional en $\mathbb{R}^2$ desde el origen hasta un punto en el plano xy con coordenadas dadas por la parte real e imaginaria del número complejo. Consulte la Figura 4.6.

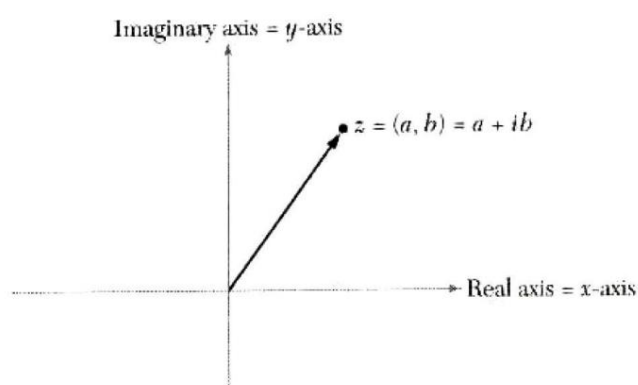


Figura 2.1: Representación vectorial de $z = a + ib$ con $\text{Re}z = a$ e $\text{Im}z = b$.

Coordenada Polar y Representación Exponencial. Escribimos un número complejo en la forma de coordenadas polares tomando la longitud r del vector $(a, b) = a + ib$ por $r = \sqrt{a^2 + b^2}$. Entonces, suponemos que el vector forma un ángulo positivo θ con el eje real donde $0 \leq \theta < 2\pi$. Consulte la Figura 2.2.

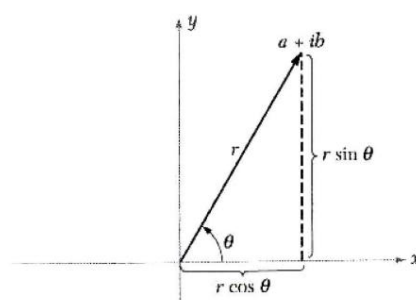


Figura 2.2: Representación en coordenadas polares del número complejo $z = a + ib$ con longitud r y ángulo θ .

Por SOHCAHTOA, obtenemos que $\tan \theta = \frac{b}{a}$ donde $a = r \cos \theta$ y $b = r \sin \theta$, por lo tanto tenemos,²

$$z = a + ib = r \cos \theta + i(r \sin \theta) = r(\cos \theta + i \sin \theta) = re^{i\theta} \quad (2.6)$$

²Derivamos la forma exponencial usando la fórmula de Euler, $e^{i\theta} = \cos \theta + i \sin \theta$.

2.4 Propiedades de los números complejos

En esta sección, presentaremos todas las propiedades básicas e importantes de los números complejos necesarias para la comprensión de las propiedades correspondientes de los cuaterniones.

2.4.1 Norma y Argumento de $z \in \mathbb{C}$

En la Figura 2.2, el vector que representa el número complejo $z = a + ib$ tiene una longitud r que también se puede denotar como $|z|$ y se llama la norma de z . El cálculo formal viene dado por:

$$|z| = \sqrt{a^2 + b^2} \quad (2.7)$$

El ángulo θ se llama argumento de z y se denota por $\arg z = \theta$. El cálculo formal viene dado por:

$$\theta = \tan^{-1} \frac{b}{a} \quad (2.8)$$

2.4.2 Complejo Conjugado de $z \in \mathbb{C}$

Dado un número complejo $z = a + ib$, entonces $\bar{z}$, que se llama complejo conjugado de z , se define como $\bar{z} = a - ib$. Esto se puede representar geométricamente como un reflejo en el eje real x que se ilustra en la Figura 2.3.

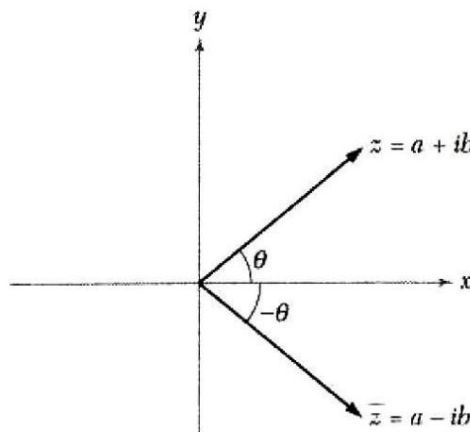


Figura 2.3: Conjugación compleja.

Por lo tanto, en forma de coordenadas polares, el conjugado complejo también se puede representar usando la misma magnitud pero ángulo opuesto.

2.4.3 Reglas algebraicas para números complejos

En esta sección, mostraremos que todas las reglas algebraicas habituales en la manipulación de números reales también se pueden aplicar a números complejos.

Reglas de adición

Los números complejos z y w satisfacen las siguientes reglas de suma:

1. $z + w = w + z$
2. $z + (w + s) = (z + w) + s$
3. $z + 0 = z$
4. $z + (-z) = 0$

Reglas de multiplicación

Los números complejos z y w satisfacen las siguientes reglas de multiplicación:

1. $zw = wz$
2. $(zw)s = z(ws)$
3. $1z = z$
4. $z(z^{-1}) = 1$ para $z \neq 0$

Podemos resolver para el inverso multiplicativo z^{-1} de $z = a + ib$ por:

$$z^{-1} = \frac{a}{a^2 + b^2} + i \frac{-b}{a^2 + b^2} \quad \text{y} \quad \overline{z^{-1}} = \frac{-b}{a^2 + b^2} + i \frac{a}{a^2 + b^2} \quad (2.9)$$

Por tanto, para $z = a + ib \neq 0$, podemos escribir:

$$z^{-1} = \frac{a}{a^2 + b^2} - i \frac{b}{a^2 + b^2} \quad (2.10)$$

Ley distributiva

Los números complejos z , w y s satisfacen las siguientes reglas de distribución,

$$z(w + s) = zw + zs \quad (2.11)$$

Es importante señalar aquí que, al igual que los números reales, los números complejos también satisfacen las propiedades asociativas, conmutativas y distributivas de la suma y la multiplicación. Compararemos este caso con el caso de las operaciones de cuaterniones que se discutirá en el próximo capítulo.

2.4.4 Suma de Números Complejos

Podemos ilustrar la suma de números complejos como la suma de vectores bidimensionales que representan los dos números complejos. Esta operación se ilustra en la Figura 2.4.

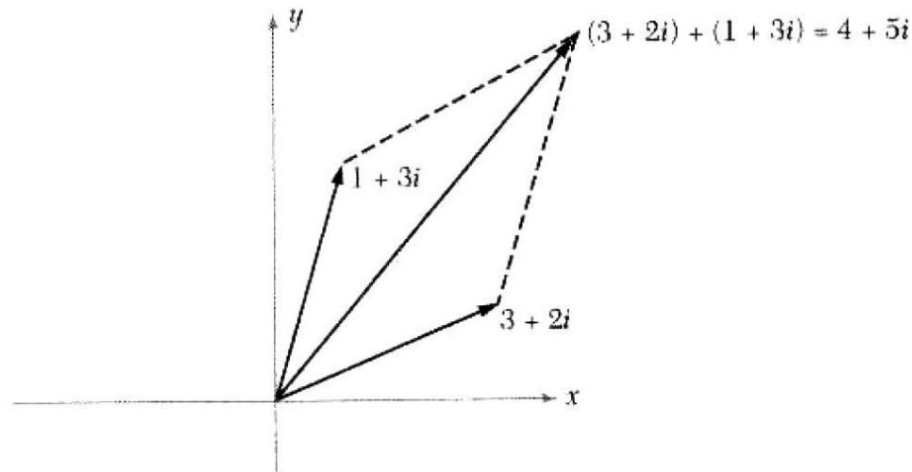


Figura 2.4: Suma de números complejos $3 + 2i$ y $1 + 3i$.

La misma representación también se puede usar para la resta de números complejos donde también se puede usar la suma sumando el primer vector al reflejo en el origen del otro vector.

2.4.5 Multiplicación de Números Complejos

Representar un número complejo en coordenadas polares o en forma exponencial ayuda a comprender el producto de dos números complejos en el sentido geométrico.

Si hacemos que $z_1 = r_1(\cos \theta_1 + i \sin \theta_1)$ y $z_2 = r_2(\cos \theta_2 + i \sin \theta_2)$, tenemos,

$$\begin{aligned} z_1 z_2 &= r_1(\cos \theta_1 + i \sin \theta_1) \cdot r_2(\cos \theta_2 + i \sin \theta_2) = \\ &= r_1 r_2 [(\cos \theta_1 \cdot \cos \theta_2 - \sin \theta_1 \cdot \sin \theta_2) + i(\cos \theta_1 \cdot \sin \theta_2 + \cos \theta_2 \cdot \sin \theta_1)] \\ &= r_1 r_2 [\cos(\theta_1 + \theta_2) + i \sin(\theta_1 + \theta_2)] = r_1 r_2 e^{i(\theta_1 + \theta_2)} \quad (2.12) \end{aligned}$$

Tenga en cuenta que aplicamos las siguientes fórmulas trigonométricas para la suma y diferencia de los productos de los cosenos y senos de los ángulos:

$$\cos(\theta_1 + \theta_2) = \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 \quad (2.13)$$

$$\sin(\theta_1 + \theta_2) = \cos \theta_1 \sin \theta_2 + \cos \theta_2 \sin \theta_1 \quad (2.14)$$

2.4. Propiedades de los Números Complejos

Proposición. Para cualquier número complejo z_1 y z_2 , proponemos que,

$$|z_1 z_2| = |z_1| \cdot |z_2| \text{ y } \arg(z_1 z_2) = \arg z_1 + \arg z_2 \pmod{2\pi} \quad (2.15)$$

Esto significa que el producto de dos números complejos es un número complejo que tiene una longitud igual a los productos de las longitudes de los dos números complejos que se multiplicaron y tiene un argumento igual a la suma de los argumentos de esos dos números complejos. Este producto se puede ilustrar en la Figura 2.5 y un ejemplo en 2.6.

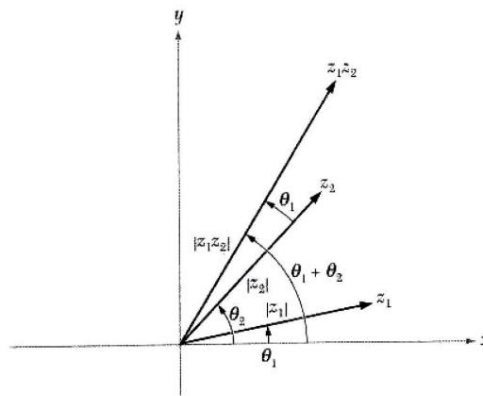


Figura 2.5: Multiplicación de números complejos z_1 y z_2 .

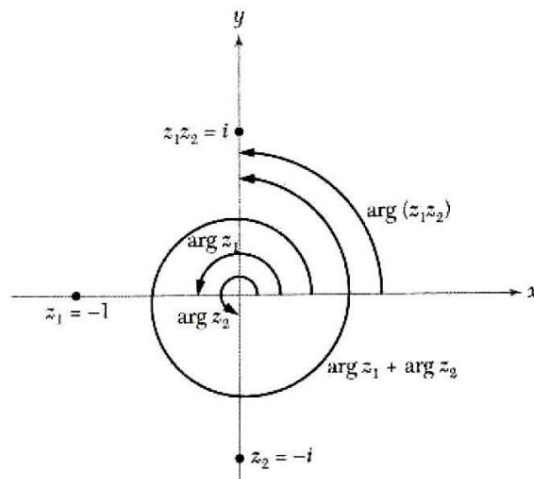


Figura 2.6: Multiplicación de números complejos -1 y $-i$.

2.5 Rotaciones en el Plano Complejo

La multiplicación de números complejos también se puede analizar mediante una aplicación ψ_z que representa la multiplicación por z tal que $\psi_z : \mathbb{C} \rightarrow \mathbb{C}$ y está definida por $\psi_z(w) = wz$. Por la proposición, el efecto de este mapa es rotar un número complejo w un ángulo que es igual a $\arg z$ en la dirección contraria a las manecillas del reloj y estirar su longitud por el factor $|z|$. Esto se puede ver en la Figura 2.7 y también en el ejemplo anterior de la Figura 2.6.

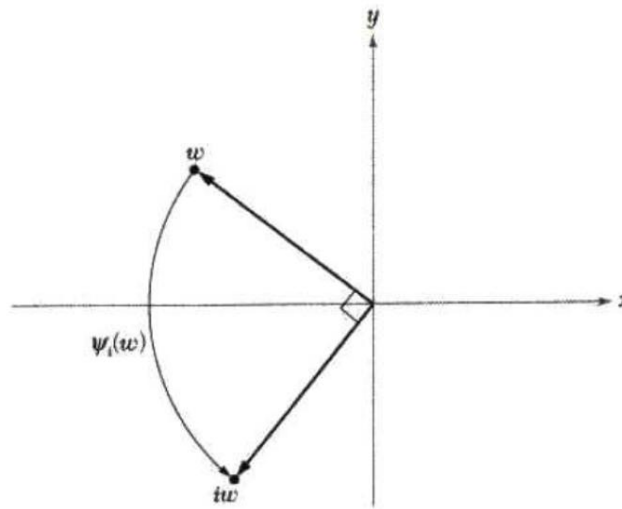


Figura 2.7: Multiplicación de wz donde $z = i$.

En el próximo capítulo, comenzaremos con la discusión de los cuaterniones y cómo su multiplicación también puede representar rotaciones en tres dimensiones.

Capítulo 3

cuaterniones

Este capítulo se ocupa de la discusión principal de los cuaterniones y sus propiedades básicas. Comenzamos detallando el dramático descubrimiento de los cuaterniones y cómo ha sacudido el mundo de las matemáticas durante su tiempo. Luego, procederemos con la definición y las propiedades importantes de los cuaterniones que nos dará una comprensión básica de este sistema numérico. también es importante para mostrar cómo los cuaterniones pueden representar rotaciones en tres dimensiones y cómo se relaciona con otras representaciones de rotación. Por último, estaremos discutiendo sobre los cuaterniones duales y su uso en robótica.

3.1 Historia de los cuaterniones

Como hemos visto en el capítulo anterior, los números complejos modelan el espacio bidimensional al tener una parte real e imaginaria que corresponde al plano xy en el que la multiplicación y suma de estos números todavía ocurrir en el mismo espacio. Durante años, el matemático irlandés Willian Rowan Hamilton había estado tratando de modelar un espacio tridimensional que tiene esta misma estructura. Estaba tratando de representarlo con un real y bidimensional parte imaginaria pero no satisfacía la estructura que quería lograr. Su hijo, William Edward Hamilton, a menudo le preguntaba sobre su progreso. en la multiplicación de tripletes a lo que respondería que sólo puede sumar o restarlos pero nunca multiplicarlos. [24]

El acto más famoso de vandalismo matemático. el avance llegó el día 16 de octubre de 1843 mientras Hamilton caminaba por el Brougham Bridge (ahora Broom Bridge) en Dublín. Tuvo un destello de perspicacia y no pudo evitar tallar en el puente las siguientes ecuaciones:

$$i^2 = j^2 = k^2 = ij = -1 \quad (3.1)$$

Capítulo 3. Cuaterniones

En su diario [10], Hamilton detalló este momento con el siguiente escrito:

“El día 16 de octubre, que resultó ser un lunes y un día del Consejo de la Real Academia Irlandesa, entraba para asistir a una presidencia, y tu madre caminaba conmigo, a lo largo del Canal Real, al que ella tal vez había conducido; y aunque ella hablaba conmigo de vez en cuando, sin embargo, una corriente subterránea de pensamiento pasaba por mi mente, lo que finalmente dio un resultado, del cual no es exagerado decir que sentí de inmediato la importancia. Un circuito eléctrico pareció cerrarse; y una chispa brilló, el heraldo (como prevé, inmediatamente) de muchos largos años por venir de pensamiento y trabajo definitivamente dirigidos, por mí mismo si se me perdonara, y en todo caso por parte de los demás, si se me permitiera siquiera hacerlo. vivir lo suficientemente claramente como para comunicar el descubrimiento. Tampoco pude resistir el impulso -por poco filosófico que haya sido- de cortar con un cuchillo en una piedra del puente Brougham, al pasar, la fórmula fundamental con los símbolos i, j, k ; es decir, $i^2 = j^2 = k^2 = ijk = -1$ que contiene la Solución del Problema.”

$$i^2 = j^2 = k^2 = ijk = -1$$

Su descubrimiento de estas leyes de la multiplicación fue también un momento decisivo en la historia del álgebra. En ese momento, se dio cuenta de que se necesitaba una cuarta dimensión y luego acuñó el término cuaternión para representar el espacio real ocupado por los elementos $1, i, j$ y k sujetos a las leyes de multiplicación anteriores.

Sus tallas se han desgastado con el tiempo, pero se colocó una placa en el puente para reemplazarlo y conmemorar el descubrimiento en la historia de las matemáticas.



Figura 3.1: Un retrato de Sir William Rowan Hamilton¹

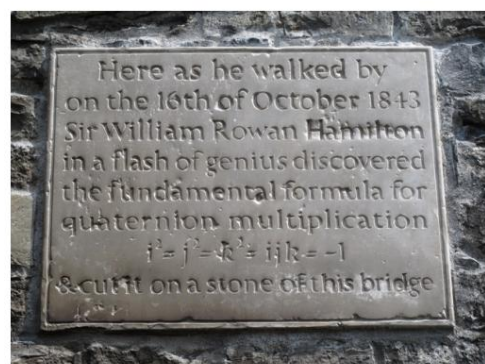


Figura 3.2: Placa conmemorativa del descubrimiento de los cuaterniones.²

3.1. Historia de los cuaterniones

12Según las palabras exactas de Hamilton [10],

“Un Cuaternión puede definirse simbólicamente como una Expresión Cuadrinomial de la forma, $q = w + ix + jy + kz$ en la que w, x, y, z son cuatro escalares o cantidades algebraicas ordinarias, mientras que i, j y k son tres nuevos símbolos que obedecen todas las leyes contenidas en la fórmula anterior y por lo tanto no están sujetos a todas las reglas usuales del álgebra.”

Hamilton definió los cuaterniones como un cociente de dos vectores. Los cuaterniones, que son un sistema de números que no satisfacen la regla conmutativa habitual de la multiplicación, inevitablemente encontraron cierta resistencia en el mundo matemático cuando fue descubierto y explicado por primera vez por Hamilton. La rivalidad entre los que prefieren los cuaterniones y los que prefieren los vectores como notaciones físicas estalló en una guerra entre los que se pusieron del lado de los cuaterniones y los que se pusieron del lado de la notación vectorial, y algunos incluso se refirieron a los cuaterniones como el mal puro.

Con la introducción de los cuaterniones, abrió nuevas posibilidades algebraicas, como el redescubrimiento de octavas u octoniones, un álgebra en octavas dimensiones, de John T. Graves realizada mediante la generalización exitosa de cuaterniones. Otro resultado bien conocido de los cuaterniones son sus generalizaciones a dimensiones superiores por parte de William Clifford, quien desarrolló una forma de construir álgebras a partir de formas cuadráticas, por lo que los cuaterniones de Hamilton surgen como esta álgebra, más conocida como Álgebra de Clifford.

Desde mediados de la década de 1880, los cuaterniones comenzaron a ser desplazados por el análisis vectorial que describía los mismos fenómenos que los cuaterniones, pero conceptualmente más simple y notacionalmente más limpio. Sin embargo, los cuaterniones tuvieron un resurgimiento desde finales del siglo XX debido a su utilidad para describir rotaciones espaciales. Esto se debe a que los cuaterniones brindan una forma elegante de describir y calcular rotaciones que es compacta y computacionalmente más eficiente y evita el bloqueo de cardán que ocurre cuando se pierde un grado de libertad cuando se aplican ciertas rotaciones. Euler Angles, en particular, es susceptible a este problema mecánico.

Por estas razones, los cuaterniones todavía se utilizan en gráficos por computadora, visión por computadora, robótica, teoría de control, procesamiento de señales, control de actitud, mecánica cuántica y muchos otros campos que implican trabajar en un espacio tridimensional.

¹https://en.wikipedia.org/wiki/William_Rowan_Hamilton

²<http://www.irelandtravelkit.com/broome-bridge-and-sir-hamiltons-eureka-moment-cabra-co-dublin/broombridge-plaque-bdolan-2010-800-etiqueta/>

3.2 El sistema numérico de cuaterniones

Comenzamos con la definición de cuaternión y sus leyes de multiplicación expandidas como también se establece en [13].

3.2.1 Definición

Los cuaterniones es el conjunto definido como,

$$H = \{w + ix + jy + kz \mid w, x, y, z \in \mathbb{R}\} \quad (3.2)$$

que es el conjunto formado por todos los números que pueden estar en la forma,

$$q = w + ix + jy + kz \text{ donde} \quad (3.3)$$

w, x, y y z son números reales e i, j y k son lo que llamamos las unidades fundamentales de cuaterniones que representan las tres dimensiones imaginarias.

3.2.2 Representaciones de cuaterniones

Forma cuaternaria estándar. Un cuaternión generalmente se representa usando su forma cuaternaria estándar escrita como,

$$q = q_0 + q_1i + q_2j + q_3k \quad (3.4)$$

en la que todas las variables corresponden a las mismas interpretaciones que en la definición.

Forma escalar y vectorial. Además de la forma estándar, un cuaternión también puede describirse como una parte escalar y una parte vectorial, donde en el caso anterior, la parte del cuaternión, $q_1i + q_2j + q_3k$, representa la parte vectorial y q_0 la parte escalar. Esto se puede representar en la llamada forma vectorial escalar escrita como,

$$q = (q_0, \mathbf{q}) \quad (3.5)$$

donde q_0 se denomina parte escalar y el vector $\mathbf{q}$ representa el vector tridimensional formado por la parte imaginaria del cuaternión y se denomina parte vectorial.

Un cuaternión sin parte real se llama cuaternión puro o cuaternión vectorial y un cuaternión sin partes imaginarias es entonces equivalente a su número real correspondiente, lo que implica que los números reales y complejos son un subconjunto de los cuaterniones. $(\mathbb{R} \subset \mathbb{C} \subset H)$

3.2. El sistema numérico de cuaterniones

3.2.3 Leyes de multiplicación de cuaterniones

Estas unidades de cuaternión i , j y k , como se indicó antes, satisfacen lo siguiente regla de multiplicación:

$$i^2 = j^2 = k^2 = -1 \quad (3.6)$$

que se expande a las siguientes leyes de multiplicación:

$$ij = k = -ji \quad (3.7)$$

$$jk = i = -kj \quad (3.8)$$

$$ki = j = -ik \quad (3.9)$$

Estas leyes se resumen en la Tabla 4.1.

$\times$	1	i	j	k
1	1	i	j	k
i		-1	k	$-j$
j		k	-1	i
k		$-j$	i	-1

Tabla 3.1: Tabla de multiplicación de cuaterniones

Estos también se pueden visualizar en el diagrama que se muestra en la Figura 3.3.³

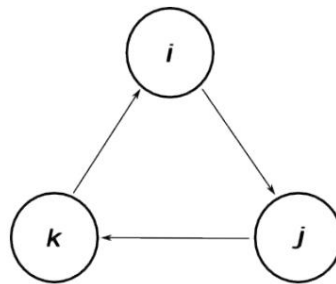


Figura 3.3: Diagrama de multiplicación de cuaterniones

Como podemos ver en la tabla y el diagrama de arriba, la multiplicación de las unidades de cuaternión no son conmutativas ($ij \neq ji$). Si la multiplicación de estas unidades van contra el flujo en el diagrama, el producto será el negativo de el producto correspondiente en el flujo normal. Esto se entenderá más claramente una vez que hemos discutido la representación de rotaciones por cuaterniones.

³<http://hforsten.com/quaternion-roots-of-modern-vector-calculus.html>

Capítulo 3. Cuaterniones

3.3 Propiedades de los cuaterniones

En esta sección, presentaremos todas las propiedades básicas e importantes de un cuaternión dadas como $q = q_0 + q_1i + q_2j + q_3k$ como también se establece en [17].

3.3.1 Conjugado de cuaterniones

El cuaternión conjugado de q se obtiene negando su parte vectorial que se muestra como,

$$q^- = q_0 - q_1i - q_2j - q_3k = (q_0, -q) \quad (3.10)$$

3.3.2 Norma de un cuaternión

La norma del cuaternión, también llamada magnitud, es la longitud del cuaternión desde el origen. Se define por:

$$\|q\| = \sqrt{qq^-} = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (3.11)$$

3.3.3 Unidad Cuaternión

Un cuaternión se llama cuaternión unitario si tiene una norma de 1, es decir,

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \quad (3.12)$$

Normalización. Cualquier cuaternión se puede normalizar para transformarlo en una unidad de cuaternión. Esto se hace dividiendo el cuaternión por su magnitud o norma, como por ejemplo:

$$\hat{q} = \frac{q}{\|q\|} = \frac{q_0 + q_1i + q_2j + q_3k}{\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}} \quad (3.13)$$

3.3.4 Inverso de un cuaternión

El inverso (recíproco) de un cuaternión se puede calcular mediante,

$$q^{-1} = \frac{q^-}{\|q\|^2} = \frac{q_0 - q_1i - q_2j - q_3k}{q_0^2 + q_1^2 + q_2^2 + q_3^2} \quad (3.14)$$

Esto se puede verificar usando la definición de la norma, ($\|q\|^2 = qq^-$),

$$qq^{-1} = q \frac{q^-}{\|q\|^2} = \frac{qq^-}{\|q\|^2} = \frac{1}{\|q\|^2} \quad (3.15)$$

3.3.5 Suma y resta de cuaterniones

La suma de cuaterniones sigue la misma regla que en los números complejos que se realiza por componentes. Si consideramos el cuaternión q dado y otro cuaternión p dado como,

$$p = p_0 + p_1i + p_2j + p_3k \quad (3.16)$$

Entonces su suma está dada por,

$$p + q = (p_0 + q_0) + (p_1 + q_1)i + (p_2 + q_2)j + (p_3 + q_3)k \quad (3.17)$$

La misma regla también se aplica a la resta de cuaterniones tales que, (3.18)

$$p - q = (p_0 - q_0) + (p_1 - q_1)i + (p_2 - q_2)j + (p_3 - q_3)k$$

3.3.6 Multiplicación y división de cuaterniones

El producto de dos cuaterniones tiene que satisfacer las leyes fundamentales que se enunciaron al principio junto con su propiedad de no conmutatividad para la multiplicación. Así, el producto de los cuaterniones p y q viene dado por,

$$\begin{aligned} pq &= (p_0 + p_1i + p_2j + p_3k)(q_0 + q_1i + q_2j + q_3k) \quad (3.19) \\ &= p_0q_0 + (p_0q_1 + p_1q_0)i + (p_0q_2 + p_2q_0)j + (p_0q_3 + p_3q_0)k \\ &\quad + (p_1q_2)ij + (p_2q_1)ji + (p_2q_3)jk + (p_3q_2)kj + (p_3q_1)ki + (p_1q_3)ik + (p_1q_1)i^2 \\ &\quad + p_0q_0 + \quad^2 + (p_2q_2)j^2 + (p_3q_3)k^2 \\ &\quad (p_0q_1 + p_1q_0)i + (p_0q_2 + p_2q_0)j + (p_0q_3 + p_3q_0)k + (p_1q_2)(k) + \\ &\quad (p_2q_1)(-k) + (p_2q_3)(i) + (p_3q_2)(-i) + (p_3q_1)(j) + (p_1q_3)(-j) + (p_1q_1)(-1) + (p_2q_2)(-1) + \\ &\quad (p_3q_3)(-1) = p_0q_0 - p_1q_1 - p_2q_2 - p_3q_3 + \\ &\quad (p_0q_1 + p_1q_0 + p_2q_3 - p_3q_2)i \\ &\quad + (p_0q_2 + p_2q_0 + p_3q_1 - p_1q_3)j + (p_0q_3 + p_3q_0 + p_1q_2 - p_2q_1)k \end{aligned}$$

Esta es una forma complicada de tomar el producto de dos cuaterniones. Hay una manera mucho más simple que consiste en usar la forma escalar y vectorial de un cuaternión y aplicar el producto punto y cruz para los vectores. Esto se hace de la siguiente manera, $pq = (p_0, p)(q_0, q) = p_0q_0$

$$- p \cdot q + p_0q + q_0p + p \times q \quad (3.20)$$

Para la división entre cuaterniones, esto se puede hacer tomando el inverso multiplicativo del divisor tal que,

$$\frac{pq}{q} = pq^{-1} \quad (3.21)$$

y aplicando el método anterior para la multiplicación de dos cuaterniones.

⁴Esta suma y multiplicación de cuaterniones se hace referencia a partir de [6]

Capítulo 3. Cuaterniones

3.3.7 Propiedades resumidas de los cuaterniones

Aquí se enumeran las propiedades resumidas de los cuaterniones.

Propiedades de adición

Los cuaterniones q , p y r satisfacen las siguientes propiedades de adición:

- Cierre: $p + q \in \mathbb{H}$
- Asociatividad: $p + (q + r) = (p + q) + r$
- Conmutatividad: $p + q = q + p$
- Identidad: $p + 0 = 0 + p = p$
- Suma: $p + q = (p_0 + q_0) + (p_1 + q_1)i + (p_2 + q_2)j + (p_3 + q_3)k$
- Diferencia: $p - q = (p_0 - q_0) + (p_1 - q_1)i + (p_2 - q_2)j + (p_3 - q_3)k$

Propiedades de multiplicación

Los cuaterniones q , p y r satisfacen las siguientes propiedades de multiplicación:

- Cierre: $pq \in \mathbb{H}$
- Asociatividad: $(pq)r = p(qr)$
- No conmutatividad: $pq \neq qp$
- Distributividad: $p(q + r) = pq + pr$
- Inversa: Si $q \neq 0$ entonces $q^{-1} \in \mathbb{H}$ tal que $qq^{-1} = q^{-1}q = 1$
- Producto: $pq = p_0q_0 - p_1 \cdot q_1 + p_0q_2 + q_0p_3 + p_1 \times q_3$
- División: $p/q = pq^{-1}$

Conjugado: $\bar{q} = q_0 - q_1i - q_2j - q_3k$

Norma: $\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2}$

Unidad Cuaternión: $\|q\| = 1$ y $p = 0, \|p\| = 0$ es un cuaternión unidad

Inversa: $q^{-1} = \frac{\bar{q}}{\|q\|^2}$ es para un cuaternión unitario u , $u\|q\|^2 = 1$

Producto escalar: $q \cdot p = q_0p_0 + q_1p_1 + q_2p_2 + q_3p_3$

3.4 Rotaciones 3D usando cuaterniones

En esta sección mostraremos cómo los cuaterniones unitarios son capaces de representar rotaciones en tres dimensiones y su relación con otras representaciones de rotaciones. ⁵

3.4.1 Representaciones de rotación 3D

El concepto de expresar una rotación en tres dimensiones como formulación matemática se extiende a la mecánica clásica, donde la cinemática rotacional es la ciencia que describe el movimiento puramente rotacional de un objeto con números.

Según el teorema de rotación de Euler, un desplazamiento general de un cuerpo rígido con un punto fijo se describe mediante una rotación alrededor de un eje. Además, esta rotación puede describirse de forma única mediante un mínimo de tres parámetros. Hay varias formas de representar rotaciones 3D y algunas necesitan más de tres parámetros, incluso si todas ellas solo tienen tres grados de libertad. Las representaciones de rotación más utilizadas se detallan aquí y se analizan cada una.

Representación de matriz de rotación

Sabemos que las rotaciones bidimensionales se pueden representar mediante la matriz,

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (3.22)$$

donde θ es el ángulo de rotación sobre el origen. Tener un solo parámetro de ángulo también implica que las rotaciones del plano solo tienen un grado de libertad.

En tres dimensiones, estamos tratando con rotaciones en los ejes principales x, y y z.

Rotar sobre el eje x corresponde a una rotación en el plano yz.

Por otro lado, girar sobre el eje y corresponde a una rotación en el plano xz y girar sobre el eje z corresponde a una rotación en el plano xy.

Estas rotaciones están representadas por las siguientes matrices,

$$\text{prescripción} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \quad (3.23)$$

⁵Esta subsección es una compilación de contenidos de las siguientes fuentes: [5][6][8][9][17]

Capítulo 3. Cuaterniones

$$R_y = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \quad (3.24)$$

$$R_z = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.25)$$

En general, si consideramos un cuerpo rígido con tres vectores unitarios ortogonales fijados a él (también denominado base). Especificar las coordenadas de esta base en su posición rotada actual en términos de los ejes de coordenadas no rotados de referencia describirá la rotación que se le aplica. Estos tres vectores unitarios r_1 , r_2 y r_3 , que constan de 3 coordenadas cada uno, dan un total de 9 parámetros. Estos parámetros forman una matriz de 3×3 que se denomina matriz de rotación.

Los elementos de una matriz de rotación se definen como sigue:

$$R = [r_1 \ r_2 \ r_3] \quad (3.26)$$

$$= \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \quad (3.27)$$

Una matriz de rotación tiene las siguientes propiedades especiales:

1. Las columnas de una matriz de rotación son vectores unitarios ortogonales.
2. Las filas de una matriz de rotación también son vectores unitarios ortogonales.
3. La transpuesta de una matriz de rotación es su inversa, $R^T = R^{-1}$. De este modo, $R^T R = R R^T = I$.
4. El determinante de una matriz de rotación es +1.

Se logra una rotación tridimensional general al combinar estas matrices de rotación en una sola matriz al realizar la multiplicación de matrices. Dado que la multiplicación de matrices no es conmutativa, es importante tener en cuenta el orden de las rotaciones, como por ejemplo,

$$R_x R_y R_z = R_z R_y R_x \quad (3.28)$$

Esto implica que rotar primero sobre el eje x, luego sobre el eje y y luego sobre el eje z no produce la misma rotación que rotar primero sobre el eje z, luego sobre el eje y y luego sobre el eje x. Esta propiedad concuerda en que las rotaciones tampoco son conmutativas. La matriz final tendrá tres parámetros de ángulo de cada matriz de rotación que corresponde a tener tres grados de libertad (estos parámetros de ángulo a menudo se denominan ángulos de Euler $[\phi, \theta, \psi]$).

3.4. Rotaciones 3D usando cuaterniones

Realizar una multiplicación matriz-vector entre la matriz de rotación y un vector en el espacio da como resultado una rotación de ese vector en el espacio mientras se conserva su longitud. Esto se puede expresar como:

$${}^Rz = Rz \quad (3.29)$$

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

donde z' es el vector z rotado por la matriz de rotación R .

La capacidad de combinar varias rotaciones en una sola matriz y la facilidad de aplicar la rotación a un vector hace que la matriz de rotación sea una forma útil y popular de representar rotaciones, incluso si es menos precisa que otras representaciones.

Representación de los ángulos de Euler

Euler Angles es históricamente la representación de rotación más popular donde una rotación general se describe como una secuencia de rotaciones alrededor de tres ejes de coordenadas ortogonales, a saber, los ejes x , y y z . La idea detrás de los ángulos de Euler es dividir una rotación en tres rotaciones más simples $[\phi, \theta, \psi]$ correspondientes a las rotaciones en los ejes principales y comúnmente denominadas cabeceo, guiñada y balanceo.

La definición de los ángulos de Euler no es única y las convenciones dependen del marco de referencia, los ejes y la secuencia sobre cómo se aplican las rotaciones. Por lo tanto, para un marco de referencia global, una rotación de ϕ en el eje x seguida de una rotación de θ en el eje y fijo, y luego una rotación de ψ en el eje z fijo se puede representar en la matriz de rotación forma premultiplicando las rotaciones sucesivas (para el marco de referencia local, se realiza la multiplicación posterior). Esto se puede expresar como: $R_{xyz} = R_z(\psi)R_y(\theta)R_x(\phi)$ $\cos \psi - \sin \psi 0$

$$= \begin{pmatrix} \cos \theta \cos \phi - \sin \theta \sin \phi \cos \psi & \cos \theta \sin \phi - \sin \theta \cos \phi \cos \psi & \sin \theta \cos \psi \\ \sin \theta \cos \phi - \cos \theta \sin \phi \cos \psi & \sin \theta \sin \phi + \cos \theta \cos \phi \cos \psi & -\sin \theta \sin \psi \\ \sin \phi \cos \psi & \cos \phi \cos \psi & \sin \psi \end{pmatrix} \quad (3.30)$$

Los ángulos de Euler se convirtieron en la forma más común de representar la rotación de un cuerpo rígido en un espacio tridimensional, ya que es la más fácil de usar y comprender. Sin embargo, las principales desventajas de los ángulos de Euler, como ser susceptibles al bloqueo de cardán y la baja precisión en la integración de cambios incrementales con el tiempo, llevaron a los investigadores a utilizar otras representaciones de rotación, como cuaterniones unitarios, como una forma de representar rotaciones de manera efectiva.

Capítulo 3. Cuaterniones

Representación eje-ángulo

El teorema de rotación de Euler implica que cualquier rotación puede expresarse como una sola rotación alrededor de algún eje. Por lo tanto, una expresión más generalizada de una rotación es la representación del eje-ángulo que se expresa simplemente como,

$$\text{rot}(\mathbf{n}, \theta), \text{ donde } \mathbf{n} = n_x \mathbf{i} + n_y \mathbf{j} + n_z \mathbf{k} \quad \text{y } \theta \in [0, \pi] \quad (3.31)$$

El eje de rotación $\mathbf{n}$ se define como cualquier vector unitario en el espacio tridimensional que no cambia con la rotación, y el ángulo θ representa la cantidad de rotación alrededor de este eje que sigue la regla de la mano derecha para determinar la dirección positiva de la rotación.

Para rotar un vector $\mathbf{z}$ a $\mathbf{z}'$ usando la representación eje-ángulo, aplicamos el Fórmula de rotación de Rodrigues definida como,

$$\mathbf{z}' = \mathbf{z} \cos \theta + \mathbf{n}(\mathbf{n} \cdot \mathbf{z})(1 - \cos \theta) + (\mathbf{z} \times \mathbf{n}) \sin \theta \quad (3.32)$$

Tenga en cuenta que si el eje no es uno de los ejes principales, suele ser difícil identificar el eje de rotación. Además, combinar dos rotaciones sucesivas representadas por un eje-ángulo no es sencillo y es mejor emplear la matriz de rotación o la multiplicación de cuaterniones para obtener la rotación final y luego convertirla a la representación del eje-ángulo.

Unidad Cuaternión Representación

Un cuaternión unidad $q = q_0 + q_1 \mathbf{i} + q_2 \mathbf{j} + q_3 \mathbf{k}$ se define como un cuaternión con norma unitaria tal que,

$$\|q\| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1 \quad (3.33)$$

La representación de una rotación por un cuaternión unitario generalmente se realiza convirtiendo otras representaciones de rotación en cuaterniones y normalizándolas. Para una representación de eje-ángulo $\text{rot}(\mathbf{n}, \theta)$ donde el eje de rotación viene dado por un vector unitario $\mathbf{n}$ y el ángulo de rotación viene dado por θ , su representación de cuaternión equivalente se formula como,

$$q = \cos \frac{\theta}{2} + \sin \frac{\theta}{2} (\mathbf{n} \cdot \mathbf{i} + \mathbf{n} \cdot \mathbf{j} + \mathbf{n} \cdot \mathbf{k}) \quad (3.34)$$

Es importante notar que q y $-q$ son equivalentes a la misma rotación incluso si sus representaciones de eje-ángulo son diferentes. Esto se debe al hecho de que girar un ángulo θ alrededor de un eje $\mathbf{n}$ es lo mismo que girar un ángulo de $2\pi - \theta$ alrededor del eje $-\mathbf{n}$.

3.4. Rotaciones 3D usando cuaterniones

Para secuencias de rotaciones que están representadas por múltiples cuaterniones, derivar la rotación final del cuaternión es tan simple como con las matrices de rotación.

Si nos dan una secuencia de rotaciones de cuaterniones $q_1, q_2, \dots, q_n$, la rotación final q_F se calcula simplemente obteniendo el producto de los cuaterniones anotando el orden de rotación y el tipo de marco de referencia.

Así, para un marco de referencia local, tenemos,

$$q_F = q_1 q_2 \dots q_n \quad (3.35)$$

Para rotar un vector tridimensional p a p' por q , el vector inicial p se representa primero como un cuaternión puro que tiene solo su parte vectorial como parte imaginaria y parte real cero. Para obtener p' se aplica lo siguiente al vector inicial p :

$$p' = qpq^{-1}, \text{ donde } q^{-1} \text{ es el inverso de } q \quad (3.36)$$

Como ejemplo, tenemos la siguiente rotación ilustrada en la Figura 3.4 donde queremos rotar el vector que representa el punto $(2, 1, 1)$ a lo largo del eje z , $\text{rot}(z, 90^\circ)$. Luego, el vector rotado se las determina usando la fórmula anterior por 90° y aplicando las reglas de multiplicación de unidades de cuaternión i, j y k . Esto dará lugar a otro cuaternión puro que luego representa el vector rotado.

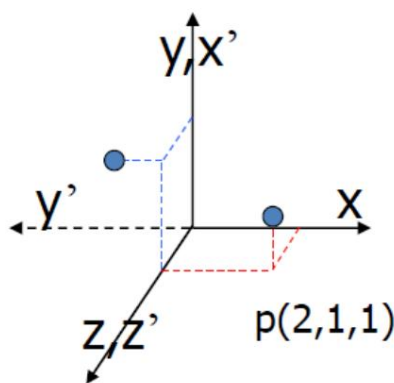


Figura 3.4: Rotación del vector $(2, 1, 1)$ de [22]

$$p = 0 + 2i + 1j + 1k = 2i + j + k \quad (3.37)$$

$$q = \text{porque } 90^\circ \text{ alrededor de } z = 0 + 0i + 0j + \frac{90^\circ}{2}k = \frac{\sqrt{2}}{2} + \frac{\sqrt{2}}{2}k \quad (3.38)$$

$$q^{-1} = \text{porque } -90^\circ \text{ alrededor de } z = 0 + 0i + 0j - \frac{90^\circ}{2}k = \frac{\sqrt{2}}{2} - \frac{\sqrt{2}}{2}k \quad (3.39)$$

$$p' = qpq^{-1} = -i + 2j + k \quad (3.40)$$

Por lo tanto, las coordenadas del vector $(2, 1, 1)$ girado 90° a lo largo del eje z son $(-1, 2, 1)$.

Capítulo 3. Cuaterniones

3.4.2 Conversiones de cuaterniones

Al tratar con cuaterniones unitarios, es necesario aprender a convertir otras representaciones de rotación en cuaterniones unitarios y viceversa. En este estudio, restringimos nuestras conversiones solo a lo que usaremos para derivar la cinemática del robot aplicando el concepto de cuaterniones.⁶

Unidad Cuaternión = Matriz de Rotación

La conversión de un cuaternión $q = q_0 + q_1i + q_2j + q_3k$ a una matriz de rotación R viene dado por la siguiente matriz,

$$R(q) := \begin{pmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2q_1q_2 + 2q_0q_3 & 2q_1q_3 - 2q_0q_2 \\ 2q_1q_2 - 2q_0q_3 & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2q_0q_1 + 2q_2q_3 \\ 2q_1q_3 + 2q_0q_2 & 2q_0q_1 - 2q_2q_3 & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{pmatrix} \quad (3.41)$$

Cada elemento de la matriz R se denota por r_{ij} para $i, j = 1, 2, 3$.

Unidad Cuaternión = Matriz de Rotación

Para obtener el mapeo inverso, al inspeccionar la relación de los elementos de R con los coeficientes de cuaterniones (q_0, q_1, q_2, q_3) se obtendrán cuatro mapeos inversos diferentes. Algunas de estas asignaciones producirán resultados complejos.

Por lo tanto, usamos las siguientes condiciones (dependiendo de los parámetros de R) para determinar qué mapeo usar:

a) Si $r_{22} > -r_{33}$, $r_{11} > -r_{22}$, $r_{11} > -r_{33}$, entonces,

$$q(R) = \frac{1}{2} \begin{pmatrix} (1 + r_{11} + r_{22} + r_{33})^{1/2} \\ (r_{23} - r_{32})/(1 + r_{11} + r_{22} + r_{33})^{1/2} \\ (r_{31} - r_{13})/(1 + r_{11} + r_{22} + r_{33})^{1/2} \\ (r_{12} - r_{21})/(1 + r_{11} + r_{22} + r_{33})^{1/2} \end{pmatrix} \quad (3.42)$$

b) Si $r_{22} < -r_{33}$, $r_{11} > r_{22}$, $r_{11} > r_{33}$, entonces,

$$q(R) = \frac{1}{2} \begin{pmatrix} (r_{23} - r_{32})/(1 + r_{11} - r_{22} - r_{33})^{1/2} \\ (1 + r_{11} - r_{22} - r_{33})^{1/2} \\ (r_{12} + r_{21})/(1 + r_{11} - r_{22} - r_{33})^{1/2} \\ (r_{31} + r_{13})/(1 + r_{11} - r_{22} - r_{33})^{1/2} \end{pmatrix} \quad (3.43)$$

⁶En nuestra aplicación, usamos particularmente las conversiones de la matriz de rotación y la representación del eje-ángulo a la unidad de cuaterniones.

3.4. Rotaciones 3D usando cuaterniones

c) Si $r_{22} > r_{33}$, $r_{11} < r_{22}$, $r_{11} < -r_{33}$, entonces,

$$q(R) = 2^{-\frac{1}{2}} \begin{pmatrix} (r_{31} - r_{13})/(1 - r_{11} + r_{22} - r_{33}) & \frac{1}{2} \\ (r_{12} + r_{21})/(1 - r_{11} + r_{22} - r_{33}) & \frac{1}{2} \\ (1 - r_{11} + r_{22} - r_{33}) & \frac{1}{2} \\ (r_{23} + r_{32})/(1 - r_{11} + r_{22} - r_{33}) & \frac{1}{2} \end{pmatrix} \quad (3.44)$$

d) Si $r_{22} < r_{33}$, $r_{11} < -r_{22}$, $r_{11} < r_{33}$, entonces,

$$q(R) = 2^{-\frac{1}{2}} \begin{pmatrix} (r_{12} - r_{21})/(1 - r_{11} - r_{22} + r_{33}) & \frac{1}{2} \\ (r_{31} + r_{13})/(1 - r_{11} - r_{22} + r_{33}) & \frac{1}{2} \\ (r_{23} + r_{32})/(1 - r_{11} - r_{22} + r_{33}) & \frac{1}{2} \\ (1 - r_{11} - r_{22} + r_{33}) & \frac{1}{2} \end{pmatrix} \quad (3.45)$$

Elegir entre estas asignaciones ayuda a reducir la inexactitud al evitar situaciones en las que el denominador es cercano a cero.

Unidad Cuaternión = Eje-Ángulo

El cuaternión q equivalente de una rotación θ alrededor de un vector unitario n en un espacio tridimensional representado por el eje-ángulo $\text{rot}(n, \theta)$ se define como,

$$q(\theta, n) := \left(\cos\left(\frac{\theta}{2}\right), n \sin\left(\frac{\theta}{2}\right) \right) \quad (3.46)$$

Tenga en cuenta que solo estamos considerando vectores unitarios, $\|n\| = 1$, en esta conversión.

Unidad Cuaternión = Eje-Ángulo

Para el mapeo inverso de la unidad cuaternión q a la representación eje-ángulo $\text{rot}(n, \theta)$, primero consideramos $q = q_0 + q_1 i + q_2 j + q_3 k$ que es la parte vectorial de q . La representación del eje-ángulo se puede calcular mediante:

$$\theta = 2 \arccos(q_0) \quad (3.47)$$

$$n = \frac{q}{\|q\|} \quad (3.48)$$

Por lo tanto, el ángulo de rotación θ puede calcularse fácilmente a partir del primer término del cuaternión unitario dado, mientras que el eje de rotación n puede derivarse simplemente normalizando la parte vectorial (parte imaginaria) del cuaternión unitario.

3.5 Cuaterniones duales

Los cuaterniones pueden representar rotaciones tridimensionales como hemos visto en la discusión anterior. Sin embargo, cuando se trata de transformaciones en tres dimensiones, también podría haber desplazamientos o traslaciones que nuestra forma actual de cuaternión no es capaz de representar. Para esta sección, introduciremos el concepto de Cuaterniones Duales que supera este límite y se utilizará en toda la duración del estudio.

3.5.1 Definición

Los cuaterniones duales proporcionan una forma de representar tanto la rotación como la traslación en un vector de transformación. Para un cuaternión unitario q que representa la rotación apropiada y para un vector p escrito como un cuaternión puro que representa el desplazamiento correspondiente, la forma más común utilizada en robótica de un cuaternión dual viene dada por el par cuaternión-vector,⁷

$$Q(q, p) = \cos \frac{\theta}{2} + \frac{\theta}{2} \text{, pecado} \frac{\theta}{2} \langle kx, ky, kz \rangle + \langle px, py, pz \rangle \quad (3.49)$$

Para la parte del cuaternión, usamos la conversión de una representación de rotación de eje-ángulo a una unidad de cuaternión. Así, θ da el ángulo de rotación y $\langle kx, ky, kz \rangle$ denota el eje de rotación. La parte del vector es simplemente el vector de desplazamiento dado por $p = \langle px, py, pz \rangle$.

Claramente, para una transformación sin rotación y solo desplazamiento, el cuaternión dual está representado por:

$$Q(q, p) = ([1, \langle 0, 0, 0 \rangle], \langle px, py, pz \rangle) \quad (3.50)$$

donde $[1, \langle 0, 0, 0 \rangle]$ representa el cuaternión identidad de la unidad.

3.5.2 Inverso de un cuaternión dual

El inverso de un cuaternión dual $Q(q, p)$ se denota como:

$$Q^{-1} = ([q^{-1}], \langle -q^{-1} p \ q \rangle) \quad (3.51)$$

cuaternión⁻¹ denota el inverso habitual del cuaternión q . Por lo tanto, para a donde $q = [s, v]$, la parte vectorial también se puede escribir como,

$$q^{-1} \text{ parte vectorial } q = -p + [-2s(v \times (-p)) + 2v \times (v \times (-p))] \quad (3.52)$$

Tenga en cuenta que p es un cuaternión puro (parte real cero) y, por lo tanto, también es un vector.

⁷Los cuaterniones duales generalmente se definen usando números duales tales como $Q = q + 2\epsilon p$ donde $\epsilon^2 = 0$. Esto se describe claramente en [14].

3.5.3 Multiplicación de un cuaternión dual

En los cuaterniones duales, la secuencia de rotaciones y desplazamientos se representa multiplicando sus correspondientes representaciones de cuaterniones duales. Por lo tanto, para las transformaciones $Q_1(q_1, p_1)$ seguidas de $Q_2(q_2, p_2)$ para un sistema de coordenadas local, su producto de cuaterniones duales viene dado por:

$$Q_1 Q_2 = (q_1, p_1) (q_2, p_2) = ([q_1 \ q_2], \langle q_1 \ p_2 \ q_1^{-1} + p_1 \rangle) \quad (3.53)$$

donde la parte del cuaternión es simplemente la multiplicación entre los cuaterniones q_1 y q_2 y la parte del vector también se puede escribir como,

$$q_1 \ p_2 \ q_1^{-1} + p_1 = p_2 + 2s_1(v_1 \times p_2) + 2v_1 \times (v_1 \times p_2) + p_1 \quad (3.54)$$

donde $q_1 = [s_1, v_1]$ y p_2 es un cuaternión puro y también actúa como vector.

El análisis de la operación para la parte de traslación del cuaternión dual muestra que la traslación siguiente es rotada primero por la rotación anterior y la traslación rotada resultante se suma luego a la traslación actual.

3.5.4 Transformación por un cuaternión dual

Transformar un vector usando cuaterniones duales sigue casi el mismo proceso que multiplicar dos cuaterniones duales. Si tenemos la transformación final dada por el cuaternión dual,

$$QF = ([w, \langle a, b, c \rangle], \langle p_x, p_y, p_z \rangle) \quad (3.55)$$

donde $[w, \langle a, b, c \rangle] = q_F$ y $\langle p_x, p_y, p_z \rangle = p_F$, y un vector inicial a transformar dado por,

$$v = \langle v_x, v_y, v_z \rangle \quad (3.56)$$

Obtenemos el vector transformado (rotado y trasladado) v' por, $v' = \langle v'_x, v'_y, v'_z \rangle$

$$v' = (q_F \ v \ q_F^{-1}) + p_F \quad (3.57)$$

Por lo tanto, un vector se transforma usando cuaterniones duales primero girándolo usando la parte de cuaternión q_F (que representa la rotación) del cuaternión dual, y luego traduciéndolo al agregarle la parte vectorial p_F (que representa la traslación) del cuaternión dual. Este proceso es similar a cómo se aplica una transformación utilizando matrices de transformación.

En el próximo capítulo, demostraremos la aplicación de los cuaterniones duales utilizándolos en la derivación de la cinemática directa e inversa de un robot con aplicación práctica en el Universal Robot UR3.

Capítulo 4

Aplicación de cuaterniones a Robot universal UR3

El objetivo principal de este estudio es poder aplicar el concepto de cuaterniones en la manipulación de movimientos de robots. Para este estudio, trabajaremos con el Universal Robot UR3 y derivaremos su cinemática directa e inversa utilizando cuaterniones. Al final de este capítulo, se prueba y presenta una aplicación de programación informática real para la cinemática de UR3.

4.1 Robot universal - UR3

El Universal Robot UR3 es un brazo robótico colaborativo. Tiene seis grados de libertad (6DOF) y se compone de seis articulaciones giratorias, comúnmente denominadas (1) base, (2) hombro, (3) codo, (4) muñeca 1, (5) muñeca 2 y (6) muñeca 3, cada una con un rango de trabajo de 360 grados.



Figura 4.1: Robot Universal UR3

¹Las especificaciones técnicas del robot se pueden encontrar en el Apéndice A.

4.2 Cinemática directa por cuaterniones duales

Forward Kinematics es generalmente más fácil de calcular que su contraparte y lo mismo ocurre con su cálculo por Dual Quaternions. Nuestro objetivo es determinar la orientación final y la posición del efector final dados todos los ángulos de articulación. Básicamente, hay al menos dos formas de derivar la cinemática directa utilizando cuaterniones duales. Puede partir de una matriz de rotación o de una representación de eje-ángulo. Cualquiera de estos dos dará como resultado la misma rotación final pero cada uno usa diferentes coordenadas para la posición TCP inicial. En este estudio, utilizaremos el enfoque con la representación del eje-ángulo.²

4.2.1 Representaciones de transformación de articulaciones

Dada la estructura cinemática del robot UR3 en la posición cero, derivamos los cuaterniones duales que representan la transformación en cada articulación derivando primero la representación del eje-ángulo de cada articulación con respecto al marco base y estableciéndola como la primera parte de nuestro sistema dual. quaternion y la segunda parte al obtener la traducción del cuadro conjunto actual al siguiente:

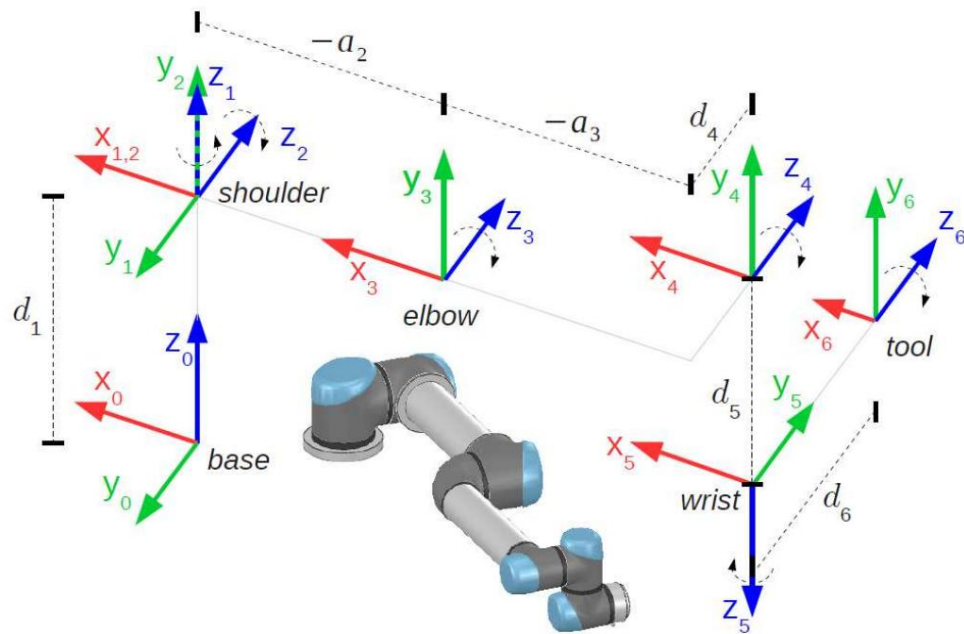


Figura 4.2: Estructura Cinemática de UR3 en posición cero ($\theta_{1,2,3,4,5,6} = 0$) [3].

²Este estudio sigue el enfoque realizado en [4] y [15]

³La convención habitual para derivar la cinemática para UR3 se puede encontrar en [3] y [11]

4.2. Cinemática directa por cuaterniones duales

Tenga en cuenta que el eje z representa el eje de rotación y estamos siguiendo el regla de la mano derecha para determinar la dirección positiva de la rotación en cada Eje de rotación. Representaremos los ejes z con respecto a la coordenada eje del marco base.

Para el UR3, también tenemos los siguientes parámetros DH:4

i	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	d_1	θ_1
2	$\alpha_1 = 90^\circ$	0	0	θ_2
3	0	a_2	0	θ_3
4	0	a_3	d_4	θ_4
5	$\alpha_4 = 90^\circ$	0	d_5	θ_5
6	$\alpha_5 = -90^\circ$	0	d_6	θ_6

Tabla 4.1: Parámetros de UR3 Denavit-Hartenberg

Para la primera articulación, en la parte de rotación, tenemos el ángulo de rotación como θ_1 y tenemos nuestro eje de rotación z_1 alineado con el eje base z_0 . Para el parte de traslación, comenzando desde el marco base hasta el marco del hombro, hay es un desplazamiento de d_1 en el eje z. Por lo tanto, nuestro cuaternión dual para el primera articulación está representada por:

$$Q_1 = \text{porque } \frac{\theta_1}{2} \text{ , pecado } \frac{\theta_1}{2} < 0, 0, 1 > , < 0, 0, d_1 >$$

Para la segunda articulación, el ángulo de rotación es θ_2 y nuestro eje de rotación z_2 es ahora paralelo al eje $-y_0$ del marco base. Para la parte de traducción, tenemos que analizar que rotar la segunda articulación resultará en una traslación en el eje x_0 y el eje z_0 que depende de a_2 y θ_2 . Por lo tanto, tenemos:

$$Q_2 = \text{porque } \frac{\theta_2}{2} \text{ , pecado } \frac{\theta_2}{2} < 0, -1, 0 > , < a_2 \cos \theta_2, 0, a_2 \sin \theta_2 >$$

Para la tercera articulación, el ángulo de rotación es θ_3 y nuestro eje de rotación z_3 , igual que z_2 , es paralelo al eje $-y_0$ del marco base. para la traducción parte, la rotación de la tercera articulación resultará en una traslación en el eje x_0 y el eje z_0 que depende de a_3 y θ_3 . Aparte de esto, para llegar a la cuarta articulación, también hay un desplazamiento de d_4 en el eje $-y_0$. De este modo:

$$Q_3 = \text{porque } \frac{\theta_3}{2} \text{ , pecado } \frac{\theta_3}{2} < 0, -1, 0 > , < a_3 \cos \theta_3, -d_4, a_3 \sin \theta_3 >$$

Los parámetros 4DH se discuten e ilustran en la Sección 4.4.1

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

Para la cuarta articulación, el ángulo de rotación es θ_4 y nuestro eje de rotación z_4 , igual que z_2 y z_3 , es paralelo al eje $-y_0$ del marco base. Para la parte de traslación, la rotación de la cuarta articulación dará como resultado una traslación en el eje x_0 y el eje z_0 que depende de d_5 y θ_5 . Por lo tanto, tenemos:

$$Q_4 = \text{porque } \frac{\theta_4}{2} \text{ , pecado } \frac{\theta_4}{2} \quad < 0, -1, 0 > \quad , < d_5 \sin \theta_4, 0, -d_5 \cos \theta_4 >$$

Para la quinta articulación, el ángulo de rotación es θ_5 y nuestro eje de rotación z_5 es paralelo al eje $-z_0$ del marco base. Para la parte de traslación, la rotación de la quinta articulación dará como resultado una traslación en el eje x_0 y el eje y_0 que depende de d_6 y θ_5 . Por lo tanto, tenemos:

$$Q_5 = \text{porque } \frac{\theta_5}{2} \text{ , pecado } \frac{\theta_5}{2} \quad < 0, 0, -1 > \quad , < -d_6 \sin \theta_5, -d_6 \cos \theta_5, 0 >$$

Para la sexta y última articulación, el ángulo de rotación es θ_6 y nuestro eje de rotación z_6 es paralelo al eje $-y_0$ del marco base. No hay traducción ya que es el último cuadro. Por lo tanto, tenemos:

$$Q_6 = \text{porque } \frac{\theta_6}{2} \text{ , pecado } \frac{\theta_6}{2} \quad < 0, -1, 0 > \quad , < 0, 0, 0 >$$

Representar las transformaciones por cuaterniones duales tiene un concepto similar con las matrices de transformación. La principal diferencia es que los cuaterniones duales son más compactos, de tal manera que solo necesitamos 7 componentes, mientras que para las matrices de transformación, usamos $4 \times 4 = 16$ componentes.

4.2.2 Cálculo de cinemática directa UR3

Después de derivar los cuaterniones duales correspondientes para cada articulación, tomamos sus productos desde el marco base hasta el marco final para obtener el cuaternión dual final que representa la transformación final y la cinemática directa del robot. Así, tenemos,

$$Q_{FK} = Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \quad (4.1)$$

Para este enfoque, las coordenadas del punto a transformar también deben ser con respecto al marco base. Por lo tanto, durante la transformación, el punto rota por la parte de rotación de Q_{FK} y luego se traslada por su parte de traslación a partir del marco base.

4.3 Cinemática inversa por cuaterniones duales

Deducir la cinemática inversa requiere más trabajo y análisis que la cinemática directa. Tendremos que aplicar el concepto de transformaciones por cuaterniones duales, lidiar con ecuaciones trigonométricas complejas y analizar geoméricamente las transformaciones realizadas por cada articulación al robot y cómo cada una de ellas afecta las coordenadas de los marcos de articulación sucesivos. Nuestro objetivo en la cinemática inversa es determinar los ángulos de articulación necesarios para mover el efector final a la posición y orientación deseadas, que se especifica mediante la transformación final, $QFK = ([w, <a, b, c>], <px, py, pz>)$.

4.3.1 Representaciones de transformaciones de articulaciones

Para comenzar con la derivación de la cinemática inversa, necesitamos los cuaterniones duales que representan la transformación realizada por cada articulación Q_i , para $1 \leq i \leq 6$, que ya hemos derivado para la cinemática directa anterior, y también sus correspondientes inversas de cuaterniones duales, Q_i^{-1}

En el futuro, usaremos las siguientes representaciones para ecuaciones largas: $\cos \theta_i \rightarrow c_i \rightarrow s_i^{-1}$, $\sin \theta_i \rightarrow s_i$, porque $\frac{\theta_i}{2} \rightarrow c_i^{-1}$, $\sin \theta_i \rightarrow s_i$

Por lo tanto, para cada articulación, tenemos:

$$\begin{aligned} Q_1 &= ([c_1 + s_1k], <d_1k>) \\ Q_2 &= ([c_2 - s_2j], <a_2c_2i + a_2s_2k>) \\ Q_3 &= ([c_3 - s_3j], <a_3c_3i - d_4j + a_3s_3k>) \\ Q_4 &= ([c_4 - s_4j], <d_5s_4i - d_5c_4k>) \\ Q_5 &= ([c_5 - s_5k], <-d_6s_5i - d_6c_5j>) \\ Q_6 &= ([c_6 - s_6j], <0>) \end{aligned}$$

y las inversas como:

$$\begin{aligned} Q_1^{-1} &= ([c_1 - s_1k], <-d_1k>) = \\ Q_2^{-1} &= ([c_2 + s_2j], <-a_2c_2i + a_2s_2k>) \\ Q_3^{-1} &= ([c_3 + s_3j], <-a_3c_3i + d_4j + a_3s_3k>) = ([c_4 \\ Q_4^{-1} &= ([c_4 + s_4j], <d_5c_4k + d_5s_4i>) \\ P_5^{-1} &= ([c_5 + s_5k], <d_6c_5j - d_6s_5i>) = \\ P_6^{-1} &= ([c_6 + s_6j], <0>) \end{aligned}$$

Usaremos las representaciones de cuaterniones duales de ambas articulaciones para obtener la relación entre las combinaciones de transformación sucesivas.

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

4.3.2 Productos de transformación de juntas

Usando los cuaterniones duales Q_i ($1 \leq i \leq 6$), que denota la cinemática transformaciones que describe las relaciones espaciales entre el éxito marcos de coordenadas y Q_i^{-1} como sus correspondientes inversas, primero definimos el producto de cuaternión dual M_i como:

$$M_i = Q_i \quad M_{i+1} \text{ donde } 1 \leq i \leq 6 \quad (4.2)$$

Para $i = 6$, tenemos que $M_6 = Q_6$. Por lo tanto, tenemos M_i como,

$$\begin{aligned} M_6 &= Q_6 \\ M_5 &= Q_5 \quad M_6 = Q_5 \quad Q_6 \\ M_4 &= Q_4 \quad M_5 = Q_4 \quad Q_5 \quad Q_6 \\ M_3 &= Q_3 \quad M_4 = Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \\ M_2 &= Q_2 \quad M_3 = Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \\ M_1 &= Q_1 \quad M_2 = Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \end{aligned}$$

Luego, definimos otro producto de cuaternión dual N_i de la siguiente manera:

$$N_{i+1} = Q_i^{-1} \quad N_i \text{ donde } 1 \leq i \leq 6 \quad (4.3)$$

Para $i = 0$, definimos N_1 como el cuaternión de transformación final, lo que significa,

$$N_1 = [R_w, T_w] = ([w, < a, b, c >], < p_x, p_y, p_z >)$$

Por lo tanto, tenemos N_i como,

$$\begin{aligned} N_1 &= [R_w, T_w] \\ N_2 &= Q_1^{-1} \quad N_1 = Q_1^{-1} \quad [R_w, T_w] \\ N_3 &= Q_2^{-1} \quad N_2 = Q_2^{-1} \quad Q_1^{-1} \quad [R_w, T_w] \\ N_4 &= Q_3^{-1} \quad N_3 = Q_3^{-1} \quad Q_2^{-1} \quad Q_1^{-1} \quad [R_w, T_w] \\ N_5 &= Q_4^{-1} \quad N_4 = Q_4^{-1} \quad Q_3^{-1} \quad Q_2^{-1} \quad Q_1^{-1} \quad [R_w, T_w] \\ N_6 &= Q_5^{-1} \quad N_5 = Q_5^{-1} \quad Q_4^{-1} \quad Q_3^{-1} \quad Q_2^{-1} \quad Q_1^{-1} \quad [R_w, T_w] \end{aligned}$$

Claramente, podemos ver que M_1 y N_1 son equivalentes a la cinemática directa QFK. Analizando más a fondo la composición de cada una de estas transformaciones entre los productos de cuaterniones duales M y N implica que $M_i = N_i$. Por lo tanto, sus términos correspondientes también deben ser iguales.

El siguiente paso es igualar cada término de cuaternión dual correspondiente entre M_i y N_i que nos darán varias ecuaciones que podemos usar para calcular para la cinemática inversa.

4.3.3 Transformaciones de marcos de la base a las muñecas

El cálculo de los productos de transformación M_i y N_i donde $(a \leq i \leq b)$ nos llevará a $((b - a) + 1) \times 7 \times 2$ ecuaciones donde los multiplicadores representan el número de transformaciones $(b - (a - 1))$, el número de términos en el cuaternión dual (7), y tanto para M como para N correspondientemente. Igualamos los términos para cada producto de transformación correspondiente y obtenemos $((b - a) + 1) \times 7$ ecuaciones y tratamos de encontrar una relación entre algunas de estas ecuaciones y manipularlas para poder resolver explícitamente los ángulos conjuntos.

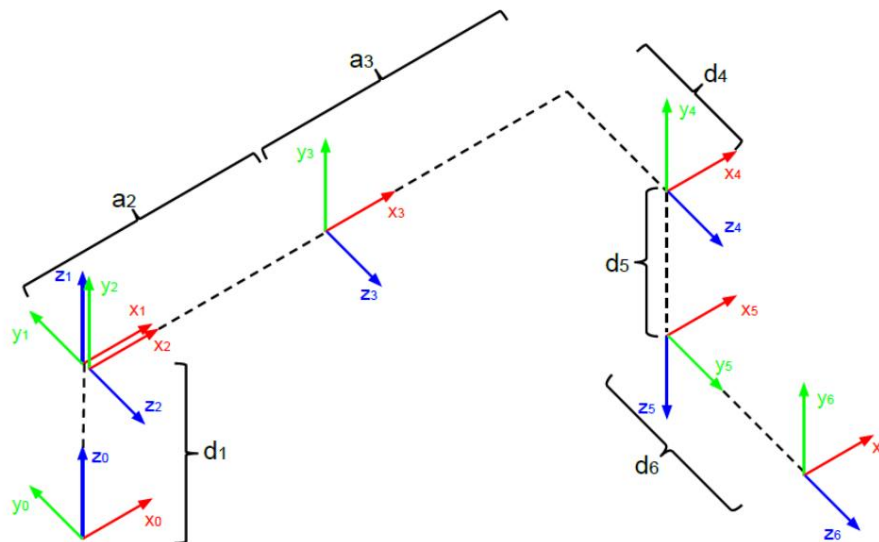


Figura 4.3: Marcos de coordenadas de UR3 ($\theta_{1,2,3,4,5,6} = 0$) de [11].

Transformación del cuadro 1 (base) al cuadro 6 (muñeca 3)

Siguiendo [4] y [15], deberíamos empezar primero con la Transformación de la base a la muñeca 3. Esto nos llevará a 84 ecuaciones donde $(1 \leq i \leq 6)$. Igualamos los productos y términos de transformación correspondientes entre M_i y N_i .

Estas ecuaciones se enumeran en el Apéndice B. La facilidad para obtener una ecuación directa para calcular los ángulos de las articulaciones depende de la estructura del robot. En el caso de UR3, no hay ecuaciones que puedan derivar directamente el valor de cualquier ángulo de articulación. Podemos encontrar una ecuación para θ_5 y luego θ_6 pero necesitamos θ_1 para resolverlas. Podemos aplicar métodos numéricos para estimar los valores de los ángulos de las articulaciones, pero en nuestro caso, necesitamos valores exactos. Para resolver esto, podemos incorporar el enfoque geométrico que se usa comúnmente para derivar la cinemática inversa de un robot y pasar a la siguiente transformación, que es la Transformación de la base a la muñeca 2.

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

Transformación de la estructura 1 (base) a la estructura 5 (muñeca 2)

Nuestra principal motivación para pasar a esta transformación es resolver para θ_1 . El enfoque geométrico es encontrar la relación entre los ángulos y los desplazamientos de las articulaciones. La siguiente figura muestra la relación entre el ángulo de articulación θ_1 con las coordenadas de la estructura 1 (base) y la estructura 5 (muñeca 2).

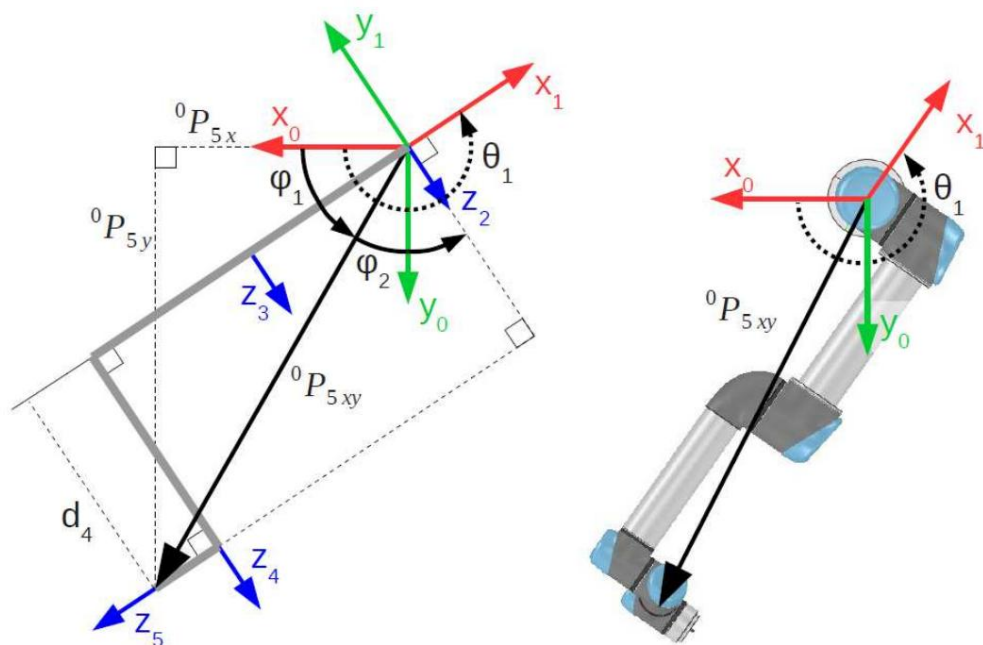


Figura 4.4: Estructura del robot UR3 del marco 1 al marco 5 (de [3])

Como podemos ver, será posible determinar el valor de θ_1 utilizando el desplazamiento entre el Marco 1, Q1 y el Marco 5, Q5. La Transformación Base a Muñeca 2 nos llevará a 70 ecuaciones donde $(1 \leq i \leq 5)$. Afortunadamente, en lugar de volver a calcular estas transformaciones, podemos usar la transformación anterior y llevar N1 a N5, ya que son simplemente equivalentes. La única diferencia es que estamos trabajando en las coordenadas del Cuadro 5. Por lo tanto, solo necesitaremos hacer una traducción de las coordenadas del Cuadro 6 al Cuadro 5. En cuanto a los productos de transformación directa, debemos comenzar con M5 en lugar de con M6. Esto se puede hacer simplemente configurando $M5 = Q5$ y la traducción a cero. Igualar los términos de M5 con sus términos correspondientes en N5 y aplicar identidades trigonométricas nos dará nuestra fórmula para θ_1 .

Esto nos permitirá luego resolver para θ_5 y θ_6 usando las ecuaciones que obtuvimos de la Transformación del Marco 1 (Base) al Marco 6 (Muñeca 3).

4.3. Cinemática inversa por cuaterniones duales

Derivación del ángulo de unión θ_1 (base)

Para derivar una fórmula para θ_1 , podemos traducir las coordenadas primero en el marco final (Muñeca 3) de vuelta al Marco 5 (Muñeca 2) y usarlas en la fórmula para extraerlas de los productos de transformación. Para volver al cuadro 5, analizamos las transformaciones de la muñeca 3 (Q6) y la muñeca 2 (Q5) y sus contribuciones a la transformación final QFK.

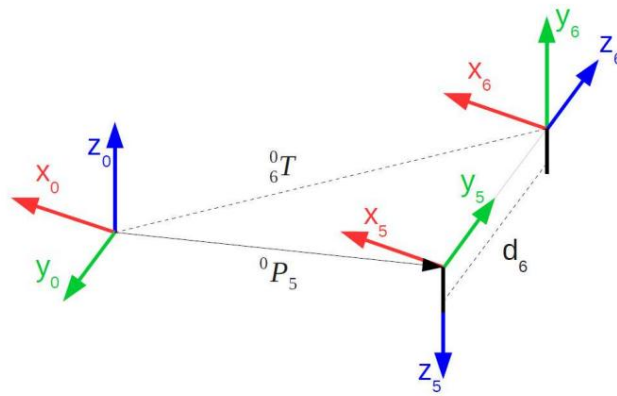


Figura 4.5: Traslación del Marco 6 (Muñeca 3) al Marco 5 (Muñeca 2)[3]

Cuadro 5 (Muñeca 2) Transformación:

$$Q5 = ([\bar{c}5 - s\bar{5}k], < -d6s5i - d6c5j >)$$

Cuadro 6 (Muñeca 3) Transformación:

$$Q6 = ([\bar{c}6 - s\bar{6}j], < 0 >)$$

Transformación final: Cuadro 1 (Base) a Cuadro 6 (Muñeca 3)

$$QFK = Q1 \quad Q2 \quad Q3 \quad Q4 \quad Q5 \quad Q6 = Q1 \quad 6$$

Claramente, no hay traslaciones realizadas por Q6, solo rotación. Por lo tanto, nuestra posición de efector final ya está determinada en la transformación hasta Q5. Así, para la parte de traducción (p) de QFK, tenemos,

$$\begin{aligned} (QFK)p &= (Q1 \quad 6)p = (Q1 \quad 5)p \\ &= (Q1 \quad Q2 \quad Q3 \quad Q4 \quad Q5)p \end{aligned} \quad (4.4)$$

Tenga en cuenta que lo que necesitamos son las coordenadas en el Cuadro 5 y solo estamos tratando con la parte de traducción del cuaternión dual.

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

Necesitamos revertir la traducción hecha por Q5. Hacemos esto usando su inversa,

$$p_5^{-1} = ([-c_5 + -s_5k], <d_5i>)$$

Por lo tanto,

$$\begin{aligned} (QFK \quad Q^{-1}_5)p &= (Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q^{-1}_5)p \\ &= (Q_1 \quad Q_2 \quad Q_3 \quad Q_4)p \\ &= (Q_1 \quad 4)p \end{aligned} \quad (4.5)$$

Todavía no tenemos un valor para θ_5 , por lo que no podemos construir completamente Q5, pero dado que solo estamos tratando con la parte de traducción, realmente no la necesitamos.

En cambio, podemos establecer un valor arbitrario para su parte de cuaternión y definir solo la parte de traducción. Recuerde que la determinación de la parte de traducción de una multiplicación de cuaterniones duales se define sin usar qB,

$$(QA \quad QB)p = <qA \quad pB \quad q \quad A^{-1} + pa> \quad (4.6)$$

Como $(QFK)p = (Q_1 \quad 5)p$, podemos tener,

$$(QFK \quad Q^{-1}_5)p = <qFK \quad p = <x_5i + y_5j \quad 5^{-1} \quad q_{FK}^{-1} + p_{FK}> \quad (4.7)$$

$$+ z_5k >= (Q_1 \quad 4)p \quad (4.8)$$

donde $<x_5, y_5, z_5>$ son las coordenadas en el Marco 5 con respecto al Marco 1.

Ahora, para calcular θ_1 , necesitamos los productos de transformación de cuaterniones duales para la transformación de la base a la muñeca 2. Como hemos dicho en nuestro análisis, podemos simplemente usar (B.13) $M_{56} = N_{56}$, donde se excluye el término $\cos \theta_5$.

Por lo tanto,

$$\begin{aligned} 0 &= y \cos \theta_1 - x \sin \theta_1 + d_4 y \cos \\ \theta_1 - x \sin \theta_1 &= -d_4 \end{aligned} \quad (4.9)$$

Aplicamos la siguiente identidad trigonométrica,

$$a \sin \theta + b \cos \theta = c$$

$$\theta = \text{atan2}(a, b) \pm \text{atan2}\left(\sqrt{a^2 + b^2 - c^2}, c\right) \quad (4.10)$$

Y ahora usamos las coordenadas del Cuadro 5 para derivar la fórmula para θ_1 ,

$$\theta_1 = \text{atan2}(-x_5, y_5) \pm \text{atan2}\left(x_5^2 + y_5^2 - \frac{d_4^2}{4}, -d_4\right) \quad (4.11)$$

Consideramos dos valores para θ_1 correspondientes a la rotación de la articulación base de establecer el Hombro a la izquierda o a la derecha.

4.3. Cinemática inversa por cuaterniones duales

Derivación del ángulo articular θ_5 (Muñeca 2)

Ahora que tenemos θ_1 , podemos usar el mismo par de ecuaciones del sistema dual productos de transformación de cuaterniones para la transformación de la base a la muñeca 3 (B.13) pero sin excluir ningún término para derivar la ecuación para θ_5 ,

$$\begin{aligned} \underline{M56 = N56} \\ -d_6 \cos \theta_5 = y \cos \theta_1 - x \sin \theta_1 + d_4 \end{aligned} \quad (4.12)$$

Podemos calcular directamente para θ_5 mediante,

$$\theta_5 = \pm \cos^{-1} \frac{x_6 \sin \theta_1 - y_6 \cos \theta_1 - d_4}{d_6} \quad (4.13)$$

Consideramos dos valores para θ_5 para la rotación de la articulación de la muñeca 2 correspondiente a la muñeca que está hacia arriba o hacia abajo.

Derivación del ángulo articular θ_6 (Muñeca 3)

Ahora tenemos θ_1 y θ_5 , entonces podemos usar los pares de ecuaciones (B.29), (B.30), (B.31) y (B.32) también de los productos de transformación de cuaterniones duales para la Transformación de la base a la muñeca 3 para derivar la ecuación para θ_6 ,

$$\begin{aligned} \underline{M21 = N21} \\ \text{porque } \frac{\theta_5}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} = w \text{ porque } \frac{\theta_1}{2} + c \text{ pecado } \frac{\theta_1}{2} \end{aligned} \quad (4.14)$$

$$\begin{aligned} \underline{M22 = N22} \\ \text{pecado } \frac{\theta_5}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} = \text{un porque } \frac{\theta_1}{2} + b \text{ pecado } \frac{\theta_1}{2} \end{aligned} \quad (4.15)$$

$$\begin{aligned} \underline{M23 = N23} \\ - \text{porque } \frac{\theta_5}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} = -\text{un pecado } \frac{\theta_1}{2} + b \text{ porque } \frac{\theta_1}{2} \end{aligned} \quad (4.16)$$

$$\begin{aligned} \underline{M24 = N24} \\ - \text{pecado } \frac{\theta_5}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} = -w \text{ sen } \frac{\theta_1}{2} + c \text{ porque } \frac{\theta_1}{2} \end{aligned} \quad (4.17)$$

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

Usando identidades trigonométricas (suma de ángulos), estos son equivalentes a,

$$c^-(2+3+4)c^-6 - s^-(2+3+4)s^-6 = \frac{wc^-1 + cs^-1}{c^-5} = \alpha \quad (4.18)$$

$$s^-(2+3+4)c^-6 - c^-(2+3+4)s^-6 = \frac{ac^-1 + bs^-1}{s^-5} = \beta \quad (4.19)$$

$$s^-(2+3+4)c^-6 + c^-(2+3+4)s^-6 = \frac{como^-1 - bc^-1}{c^-5} = \gamma \quad (4.20)$$

$$c^-(2+3+4)c^-6 + s^-(2+3+4)s^-6 = \frac{ws^-1 - cc^-1}{s^-5} = d \quad (4.21)$$

Multiplicando (4.18) por s^-6 nos dará,

$$c^-(2+3+4)c^-6s^-6 - s^-(2+3+4)s^-6 \frac{2}{6} = \alpha s^-6 \quad (4.22)$$

Y multiplicando (4.19) por c^-6 nos dará:

$$s^-(2+3+4)c^-6 \frac{2}{6} - c^-(2+3+4)s^-6c^-6 = \beta c^-6 \quad (4.23)$$

Obtener la suma de estas dos ecuaciones nos dará,

$$s^-(2+3+4)(c^-6 \frac{2}{6} - s^-6 \frac{2}{6}) = -s^-(2+3+4)c^-6 = \alpha s^-6 + \beta c^-6 \quad (4.24)$$

Entonces, multiplicando (4.20) por c^-6 nos dará,

$$s^-(2+3+4)c^-6 \frac{2}{6} + c^-(2+3+4)s^-6c^-6 = \gamma c^-6 \quad (4.25)$$

Y multiplicando (4.21) por $-s^-6$ nos da:

$$-c^-(2+3+4)c^-6s^-6 - s^-(2+3+4)s^-6 \frac{2}{6} = -\delta s^-6 \quad (4.26)$$

Obtener la suma de estas dos ecuaciones nos dará,

$$s^-(2+3+4)(c^-6 \frac{2}{6} - s^-6 \frac{2}{6}) = -s^-(2+3+4)c^-6 = \gamma c^-6 - \delta s^-6 \quad (4.27)$$

Podemos restar las ecuaciones (4.24) y (4.27) para obtener lo siguiente simplificado ecuación en la que solo θ_6 es indeterminado:

$$\alpha s^-6 + \beta c^-6 = \gamma c^-6 - \delta s^-6 \rightarrow s^-6(\alpha + \delta) = c^-6(\gamma - \beta)$$

Reorganizar esta ecuación para resolver θ_6 nos dará la siguiente fórmula,

$$\theta_6 = 2\arctan2(\gamma - \beta, \alpha + \delta) \quad (4.28)$$

Solo hay un valor para θ_6 que corresponde a la rotación de la articulación de la muñeca 3.

4.3.4 Transformación de marcos de hombro a muñeca

Ahora nos queda solo θ_2 , θ_3 y θ_4 por determinar. Note que sus articulaciones correspondientes forman un manipulador plano 3R como se ve en la figura abajo,

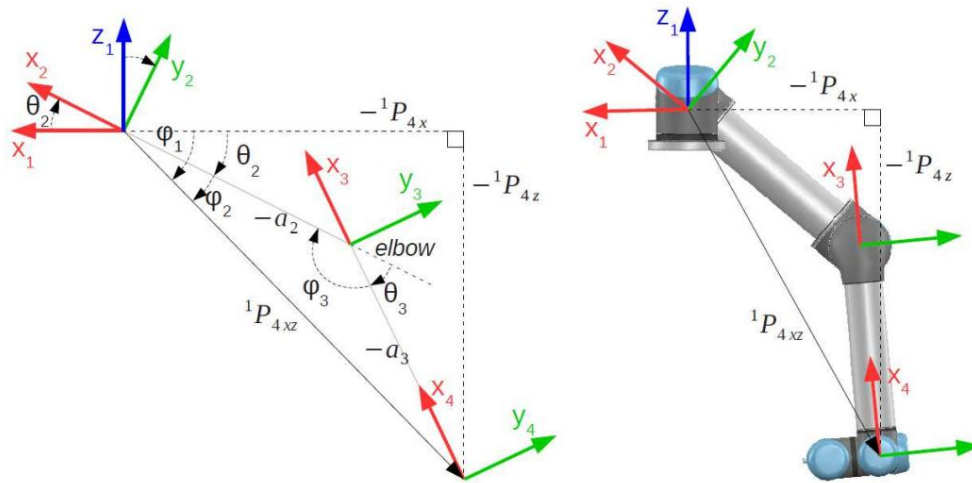


Figura 4.6: Manipulador 3R-Planar formado por UR3 Articulaciones 2 a 4 de [3].

Por lo tanto, podemos obtener los productos de transformación M_i y N_i donde ($2 \leq i \leq 4$). Esto nos llevará a 42 ecuaciones. Igualar los términos para cada producto de transformación correspondiente nos dejará con $\frac{42}{2} = 21$ ecuaciones y tratamos de encontrar una relación entre algunas de estas ecuaciones y las manipulamos para poder resolver explícitamente nuestros ángulos conjuntos restantes para que sean determinados.

Por lo tanto, tenemos la siguiente representación de cuaternión dual para cada articulación,

$$Q_2 = ([\bar{c}_2 - s_2j], < a_2c_2i + a_2s_2k >)$$

$$Q_3 = ([\bar{c}_3 - s_3j], < a_3c_3i - d_4j + a_3s_3k >)$$

$$Q_4 = ([\bar{c}_4 - s_4j], < 0 >)$$

y las inversas como:

$$Q_2^{-1} = ([\bar{c}_2 + s_2j], < -a_2c_2i >)$$

$$Q_3^{-1} = ([\bar{c}_3 + s_3j], < -a_3c_3i + d_4j >)$$

$$Q_4^{-1} = ([\bar{c}_4 + s_4j], < 0 >)$$

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

Por lo tanto, nuestro producto de transformación de cuaterniones duales M_i se formula como:

$$\begin{aligned} M_4 &= Q_4 \\ M_3 &= Q_3 \quad M_4 = Q_3 \quad Q_4 \\ M_2 &= Q_2 \quad M_3 = Q_2 \quad Q_3 \quad Q_4 \end{aligned}$$

Y el producto de cuaternión dual N_i como:

$$\begin{aligned} N_2 &= Q_2^{-1} [w_4, \langle a_4, b_4, c_4 \rangle, \langle x_4, y_4, z_4 \rangle] \\ N_3 &= Q_1^{-1} \quad N_2 = Q_1^{-1} \quad Q_2 \quad Q_4 \\ N_4 &= Q_1^{-1} \quad N_3 = Q_1^{-1} \quad Q_2^{-1} \quad Q_4 \end{aligned}$$

Estas ecuaciones se enumeran en el Apéndice C.

Aparte de esto, también necesitaremos traducir nuestras coordenadas base y coordenadas finales ya que estamos trabajando solo desde el Cuadro 2 (Hombro) al Cuadro 4 (Muñeca 1). Para traducir las coordenadas, usamos la misma técnica que usamos para determinar las coordenadas para θ_1 ,

$$Q_{FK} = Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 = Q_1 \quad Q_6 \quad (4.29)$$

$$Q_1^{-1} \quad Q_{FK} = Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6 \quad = Q_2 \quad Q_6 \quad (4.30)$$

$$Q_1^{-1} \quad Q_{FK} \quad Q_6^{-1} = Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad = Q_2 \quad Q_5 \quad (4.31)$$

$$Q_1^{-1} \quad Q_{FK} \quad Q_6^{-1} \quad Q_5^{-1} = Q_2 \quad Q_3 \quad Q_4 \quad = Q_2 \quad Q_4 \quad (4.32)$$

Tenga en cuenta que ya podemos formar completamente Q_1 , Q_5 y Q_6 . Para Q_4 , solo necesitamos revertir su traducción. Para hacer esto, usamos su inverso Q_4^{-1} de la misma manera que lo hicimos en la traducción anterior. Tenemos,

$$Q_4^{-1} = ([-c_5 + -s_5k], \langle d_6i \rangle)$$

Y revertimos la traducción por,

$$\begin{aligned} (Q_2 \quad Q_4 \quad Q_1^{-1} \quad Q_4) p &= (Q_2 \quad Q_3 \quad Q_4 \quad Q_1^{-1} \quad Q_4) p \\ &= (Q_2 \quad Q_3) p \\ &= (Q_2 \quad Q_3) p \end{aligned} \quad (4.33)$$

Obtenemos las coordenadas por la parte de traducción de este producto,

$$\begin{aligned} (Q_2 \quad Q_4 \quad Q_1^{-1} \quad Q_4) p &= \langle \frac{2}{q_4} \text{ pag}_4^{-1} \quad (q_4 \frac{2}{4})^{-1} + \text{pag}_4^2 \rangle \\ &= \langle x_4i + y_4j + z_4k \rangle \end{aligned} \quad (4.34)$$

donde $\langle x_4, y_4, z_4 \rangle$ son las coordenadas en el Marco 4 con respecto al Marco 2.

4.3. Cinemática inversa por cuaterniones duales

Derivación del ángulo articular θ_3 (codo)

Para derivar una fórmula para θ_3 , hacemos uso de las ecuaciones (C.19) y (C.21) en la Transformación Hombro a Muñeca 1 junto con la previamente calculada coordenadas $\langle x_4, y_4, z_4 \rangle$.

$$\underline{M_{25} = N_{25}}$$

$$a_3 \cos(\theta_2 + \theta_3) + a_2 \cos \theta_2 = x_4 \quad (4.35)$$

$$\underline{M_{27} = N_{27}}$$

$$a_3 \sin(\theta_2 + \theta_3) + a_2 \sin \theta_2 = z_4 \quad (4.36)$$

Elevar al cuadrado estas dos ecuaciones y sumarlas nos dará,

$$a_3^2 + 2a_2a_3 \cos \theta_3 + a_2^2 = x_4^2 + z_4^2 \quad (4.37)$$

Ahora, podemos calcular directamente para θ_3 por,

$$\theta_3 = \pm \arccos \frac{x_4^2 + z_4^2 - a_2^2 - a_3^2}{2a_2a_3} \quad (4.38)$$

Consideramos dos valores para θ_3 que corresponde a la rotación de la articulación del codo de estar arriba o abajo.

Derivación del ángulo articular θ_2 (hombro)

Como ya tenemos θ_3 , podemos calcular θ_2 usando las ecuaciones (C.12) y (C.14) también en la Transformación Hombro a Muñeca 1 usando el mismo coordenadas

$$\underline{M_{35} = N_{35}}$$

$$a_3 \cos \theta_3 = -a_2 + x_4 \cos \theta_2 + z_4 \sin \theta_2 \quad (4.39)$$

$$\underline{M_{37} = N_{37}}$$

$$a_3 \sin \theta_3 = -x_4 \sin \theta_2 + z_4 \cos \theta_2 \quad (4.40)$$

Y usando manipulaciones similares, podemos calcular directamente para θ_2 por,

$$\theta_2 = \text{atan2}(z_4(a_2 + a_3 \cos \theta_3) - x_4a_3 \sin \theta_3, x_4(a_2 + a_3 \cos \theta_3) + z_4a_3 \sin \theta_3) \quad (4.41)$$

Consideramos un valor para θ_2 que corresponde a la rotación del Hombro.

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

Derivación del ángulo articular θ_4 (Muñeca 1)

Dados θ_3 y θ_2 , podemos calcular θ_4 usando las ecuaciones (C.1) y (C.3) y la parte de rotación de la transformación Q_2 dada por $q_4 = w_4 + a_4i + b_4j + c_4k$.

$$\begin{aligned} & \underline{M_{41} = N_{41}} \\ & \text{porque } \frac{\theta_4}{2} = w_4 \text{ porque } \frac{\theta_2 + \theta_3}{2} - b_4 \text{ pecado } \frac{\theta_2 + \theta_3}{2} \end{aligned} \quad (4.42)$$

$$\begin{aligned} & \underline{M_{43} = N_{43}} \\ & - \text{pecado } \frac{\theta_4}{2} = w_4 \text{ pecado } \frac{\theta_2 + \theta_3}{2} + b_4 \text{ porque } \frac{\theta_2 + \theta_3}{2} \end{aligned} \quad (4.43)$$

Podemos calcular directamente para θ_4 mediante,

$$\theta_4 = -2 \operatorname{atan2}(w_4 \sin(\frac{\theta_2 + \theta_3}{2}) + b_4 \cos(\frac{\theta_2 + \theta_3}{2}), w_4 \cos(\frac{\theta_2 + \theta_3}{2}) - b_4 \sin(\frac{\theta_2 + \theta_3}{2})) \quad (4.44)$$

Consideramos un valor para θ_4 que corresponde a la rotación de la articulación Muñeca 1.

4.3.5 Soluciones de ángulos de unión

Usar las ecuaciones (4.11), (4.13), (4.28), (4.38), (4.41) y (4.44) nos permitirá calcular una combinación de ángulos ($\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$) que danos nuestra transformación deseada Q_1 . Cada uno de nuestros ángulos conjuntos de estas ecuaciones tiene el siguiente número de valores posibles,

$$\begin{array}{ccc} \theta_1 \times 2 & \theta_2 \times 1 & \theta_3 \times 2 \\ 4 \times 1 & \theta_5 \times 2 & \theta_6 \times 1 \end{array}$$

Por lo tanto, hay un total de $2 \times 1 \times 2 \times 1 \times 2 \times 1 = 8$ posibles combinaciones de ángulos de articulación. Todavía es necesario hacer alguna filtración de estos valores en nuestra aplicación práctica de programación porque aún puede suceder que solo algunas de estas combinaciones nos den exactamente nuestra transformación deseada. En este estudio, se hace usando el principio de cinemática directa. Ahora, tenemos nuestra entrada como los ángulos conjuntos de nuestra combinación actual y obtenemos su transformación correspondiente,

$$Q_1(\theta_1) \quad Q_2(\theta_2) \quad Q_3(\theta_3) \quad Q_4(\theta_4) \quad Q_5(\theta_5) \quad Q_6(\theta_6) = QFK \quad (4.45)$$

Verificamos si esta transformación es equivalente a nuestra transformación deseada donde también debemos considerar algunas pequeñas discrepancias debido a errores numéricos. Este procedimiento se demuestra en nuestra aplicación de programación.

4.4 Aplicación práctica de programación

La aplicación de programación Python para derivar el directo y el inverso.
La cinemática de UR3 se presenta en esta sección.

4.4.1 Parámetros de Denavit-Hartenberg

Los parámetros Denavit-Hartenberg de UR3 se dan en la siguiente tabla:

UR3				
Kinematics	theta [rad]	a [m]	d [m]	alpha [rad]
Joint 1	0	0	0.1519	$\pi/2$
Joint 2	0	-0.24365	0	0
Joint 3	0	-0.21325	0	0
Joint 4	0	0	0.11235	$\pi/2$
Joint 5	0	0	0.08535	$-\pi/2$
Joint 6	0	0	0.0819	0

Tabla 4.2: Parámetros Denavit-Hartenberg para UR3 de [20]

Estos parámetros se describen mediante el siguiente diagrama:5

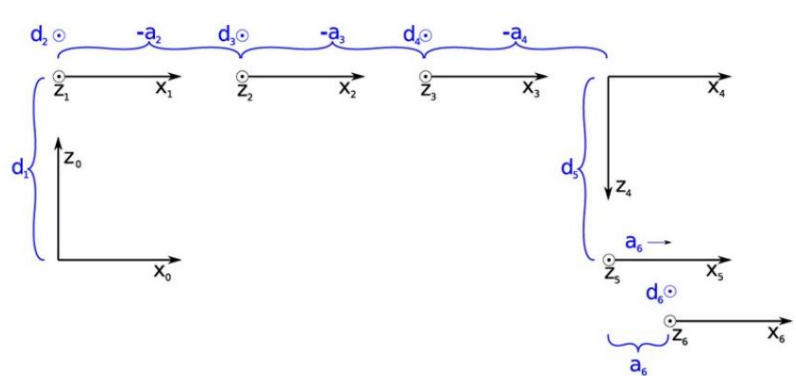


Figura 4.7: Diagrama de parámetros UR3 DH de [20]

La configuración de los parámetros DH en el programa sigue el orden en la Tabla 4.1 usando los valores dados por la Tabla 4.2.

```
1 # UR3 Denavit-Hartenberg Parámetros
2 re = [0.1519, 0.101, 0.21325, 0.0819, 0.08535, 0.0819],
3 un = [0, 0, -0.24365, -0.21325, 0, 0],
4 alfa = [0, pi/2, 0, 0, pi/2, -pi/2]
```

5El uso y la derivación de los parámetros de Denavit-Hartenberg se pueden encontrar en [23]

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

4.4.2 Aplicación de programación Python

Usamos la biblioteca urx python para comunicarnos y controlar el robot UR3.

```

1 urx de importación
2 de urx importar urrobot
3 robar = urx. robot ( " 1 4 7. 2 2 9. 1 3 2. 2 4 9 " ) #Dirección IP de UR3

```

La dirección IP del robot UR3 se puede ver en la pestaña 'Acerca de' en el UR3 Interfaz gráfica de usuario de Polyscope.

Clase de cuaternión dual

Primero definimos nuestro objeto de cuaternión dual. Como hemos descrito, consiste de dos cuaterniones donde el primer cuaternión representa la rotación y el segundo cuaternión representa la traslación. La biblioteca pyquaternion es utilizado para definir estos objetos de cuaternión y creamos una clase para el Dual Cuaternión siguiendo las propiedades que enumeramos en el capítulo de cuaternión. El la multiplicación de cuaterniones duales también se define en esta clase siguiendo el propiedades que enumeramos en el capítulo de cuaterniones duales.

```

1 clase DualQuaternion:
2 # l inicializa un Cuaternión Dual
3     def __init__(self, q=Cuaternión(), p=0):
4         self.q = q
5         self.p = p
6
7 # Multiplicación de cuaterniones duales
8     def mult(self, dq):
9         # Para la parte de rotación
10            q1 = self.q
11            q2 = dq.q
12
13        # Para la parte de traducción
14        p1 = self.p
15        p2 = dq.p
16
17        return DualQuaternion(q1 * q2, q1.p * q2.p + p1)
18
19 # Comprueba la igualdad de los cuaterniones duales
20     def equals(self, dq):
21         # Para la parte de rotación
22        q1 = self.q
23        q2 = dq.q
24
25        # Para la parte de traducción
26        p1 = self.p
27        p2 = dq.p
28

```

4.4. Aplicación práctica de programación

```

29 # Comprueba la igualdad
30 print ( "\nComparación de cuaterniones duales: " )
31 imprimir ( " Rotación : q1 , "
32 print ( " , p2 )
33
34 si ( ( np . allclose ( p1 , p2 , atol=1e-03)
35     o np . todo cerrado ( p2 , p1 , atol=1e-03) )
36     y ( np . allclose ( q1 , q2 , atol=1e-03)
37         o np . todo cerrado ( q2 , q1 , atol=1e-03) ) ) :
38     volver verdadero
39     volver F al se
40
41 # Transforma un punto en el espacio
42 def transform ( self , pose ) :
43     # Pose girada
44     ro tP ose = self . q rotar (posicionar)
45     print ( "\nDespués de la rotación: ro tP ose )
46
47     # Postura traducida
48     tran sP ose = self . pag . vector + ro tP ose
49     print ( " DESPUÉS DE LA TRADUCCIÓN : TRANSP OSE )
50
51     TRANSPORTE DE RETORNO

```

Al rotar un vector, usamos el método de rotación del pycuaternión ya que también normaliza el cuaternión antes de aplicar la rotación. Esto se hace en el siguientes códigos en la biblioteca pyquaternion:

```

1 def rotar (autovector): ,
2     Rotar un vector 3D por la rotación almacenada en el
3     Objeto de cuaternión.
4
5     Parámetros:
6     vector : Un vector de 3 vectores especificado como cualquier secuencia ordenada
7     de 3 números reales se corresponden con los valores x, y y z.
8     Algunos tipos que se reconocen son: matrices numpy
9     , listas y tuplas.
10    Un 3-v ector también puede ser representado por un
11    Objeto de cuaternión que ' sscalarpartis 0 y vectorpartis
12    el 3-v ector requerido .
13    Así es posible llamar 'Cuaternión'. rotar ( q
14    ) ' con un o el rquater ni onobject como entrada .
15
16    Devoluciones :
17    El vector rotado devolvió como el mismo tipo que era
18    especificado en la entrada.
19
20    Rai ses :
21    TypeError: si alguno de los elementos del vector no se puede
22    convertido a número real.

```

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

```

15         V alorError :      if ' vector 3-v ' no se puede interpretar como
    ectorora Quaternion object .
    """
17
18     if isinstance ( vector , Quaternion ) :
19         devolve a uno mismo. rotar cuaternión (vector)
20     q = Cuaternión ( vector=vector )
21     a = uno mismo. rotar cuaternión ( q ) . vector
22     if isinstance ( lista de vectores ) :
23         l = [ x para x en un ]
24         volver l
25     elif isinstance ( vectorl = [ x for , tu por favor ) :
26         x in a ]
27         volver por favor ( l )
28     más :
29         devolver un
30
31     def rotar cuaternión ( self , q ) :
32         Rotate aquater ni onvector usando la rotación almacenada.
33
34         Parámetros:
35         q : El vector a rotar , inquater ni en forma ( 0 +
36         xi + yj + kz )
37
38         Devoluciones :
39         Un objeto Quaternion que representa el vector rotado
40         inquater ni en de ( 0 + xi + yj + kz )
41
42         ser . norm alli ( )
43         volver yo mismo q yo . conjugado
44
45     def norm ali se ( self ) :
46         """ Se garantiza que el ob jec tis será au ni tquater ni onafter
47         llamando a esto
48         operación A MENOS QUE el objectiseq ui v al entto Quaternion
49         ( 0 ) """
50
51         si no uno mismo. esunidad ( ) :
52             n = uno mismo. norma
53             si n > 0 :
54                 ser . q = uno mismo. q / n

```

También notamos que en el código anterior, definimos cómo determinar si dos los cuaterniones duales son aproximadamente iguales. Usamos el método `allclose` de la biblioteca `numpy` que verifica si dos matrices son iguales en cuanto a elementos dentro de un tolerancia. Hacemos esto para la cinemática inversa cuando queremos filtrar las combinaciones de ángulos resultantes por sus correspondientes transformaciones y debemos considerar los errores numéricos que pueden surgir durante los cálculos.

4.4. Aplicación práctica de programación

Cinemática directa

Para la cinemática directa, nuestro objetivo es determinar la transformación final (orientación y posición) del efector final del robot dadas sus articulaciones rotaciones angulares. Primero, necesitamos obtener los ángulos de unión actuales de nuestro robot. Hacemos esto usando el método `getj` de la biblioteca `urx`.

```
1 # Obtiene los elementos de unión actuales
2 tintineos = robar.conseguir()
```

Esto devuelve el valor de los seis ángulos de las articulaciones desde la base hasta la muñeca.

3. También definimos dos enfoques para derivar un cuaternión dual. uno es por la representación eje-ángulo (AA) y por la matriz de transformación (TM). Definimos diferentes coordenadas TCP iniciales para cada uno de estos enfoques.

```
1 # Establecer la posición inicial de TCP en el espacio en función de la
   marco
2 fin TCP = np.matriz([0 3 TCP basef = np.matriz([0, 0, 2]) # basado en el marco final (0, 0, 1)
([0, -0.2, 0]) # basado en el marco base (0, -1, 0)
```

La razón de esto es por el primer enfoque que es de AA, debemos considerar las coordenadas TCP con respecto al marco Base. Y para la TM enfoque, debemos considerar las coordenadas TCP con respecto a su marco.

El método cinemática adelantada ahora llama la derivación de la doble cuaterniones para cada seis articulaciones según el enfoque elegido y calcula el cuaternión dual final que representa la transformación final al multiplicar siguiendo el orden de transformación (articulaciones).

```
1 # Obtiene la posición actual usando cinemáticas directas
2 def getforward kinematics ( angles ):
3
4     print ( "\nÁngulos de unión actuales: ", tintineos )
5
6 # Deriva las correspondientes cuaterniones duales para cada junta
   a partir de su Representación del Eje-Ángulo
7     si ( respuesta == "AA" ):
8         print ( "\nDerivar cinemáticas directas a partir del eje del ángulo
   Representación . . . " )
9         DualQuatList = derivado al quaternión AA ( angles ) -
10        poseinicial = TCP basef -
11 # Deriva las correspondientes cuaterniones duales para cada junta
   a partir de sus Matrices de Transformación
12     elif ( respuesta == "TM" ):
13         print ( "\nDerivación de cinemática directa a partir de transformación
   Matriz. . . " )
14         DualQuatList = derive Du al Quaternión de TM ( angles ) -
15         poseinicial = final TCP -
16     más :
17         print ( "¡Entrada incorrecta!" )
```

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

```

18         devolver
19
20 # Calcula el cuaternión dual final multiplicando todos los
    Dual Quaterniones
21 FKDualQuat = DualQuaternion ( )
22 para dq en DualQuatList:
23     FKDualQuat = FKDualQuat . múltiple (dq)
24
25     print ( "\nForward Kinematics (by Dual Quaternion): \n (["
        FKDualQuat.q , "], , FKDualQuat.p , ">)" )
26
27 # Transformar pose inicial por Multiplicación de Quaternion-Vector
28     print ( "\n Coordenadas TCP iniciales (a ceroposición): posición
        inicial) -
29     cursar = FKDualQuat . transformar (postura inicial) -
30     print ( "\nCoordenadas TCP transformadas: \n", np. round (currpose
        , 3 ) )
31
32     volver FKDualQuat

```

La salida de este método es la pose actual o las coordenadas TCP de el robot después de haber sido transformado (girado y trasladado) y con respecto al Marco Base.

Cuaterniones duales de la representación eje-ángulo

La derivación de la cinemática directa por el enfoque de eje-ángulo es hecho en el método derivar dual quat de AA:

```

1 # Derive el cuaternión dual a partir del ángulo y el eje de rotación
2 def derivar dual quat from AA ( jangles ) :
3 # Obtiene los ángulos de las articulaciones de la posición actual
4     theta = tintageo
5
6 # Set traducción para cada articulación
7     trans = ( [[ 0 d [ 0. ] ] , 0 [ a [ 2 ]
8                 cos ( theta [ 1 ] ) , [ a [ 3 ] 0 , a [ 2 ] sin ( theta [ 1 ] ) ] , #P1
9                 cos ( theta [ 2 ] ) , [ re -d [ 3 ] a [ 3 ] sen ( theta [ 2 ] ) ] , # P3
10                [ 4 ] sin ( theta [ 3 ] ) , [-d 0 , -d [ 4 ] cos ( theta [ 3 ] ) ] , #P4
11                [ 5 ] sin ( theta [ 4 ] ) , [ 0 -d [ 5 ] coseno ( theta [ 4 ] ) , 0 ] , #P5
12                0 ] ] , #P6
13
14 # Builds the Dual Quaternion
15     DQLista = [ ]
16     al p = 0
17
18     para mi In range ( 6 ) :
19         # Construye el dual quaternion
20         qscalar = cos ( theta [ i ] / 2 )

```

4.4. Aplicación práctica de programación

```

21     al p += alfa a [ yo ]
22     qvector = np. multiply ( sin ( theta [ i ] / 2 ) , al p ) ) [ 0 , -sen ( al p ) , porque (
23
24     quatRot = np . concatenar ( ( [ qscalar ] , qvector ) )
25
26     DualQuat = DualQuaternion ( Quaternion ( quatRot ) , Quaternion (
27     vector = trans [ yo ] ) )
28     print ( "Cuaternión dual: ( [ " , Dual Quat . q , " ] , , DualQuat . pag , ">"
29     " ) )
30
31     # Almacenar el Dual Quaternion en la lista
32     DQList. agregar (DualQuat)
33
34     devolver DQList

```

Para calcular la transformación final, primero definimos la traducción para cada articulación. Esto corresponde a la parte de traducción de nuestros cuaterniones duales. en el que usamos los parámetros de Denavit-Hartenberg (a , d , θ). Entonces nosotros construya los cuaterniones de rotación para las seis articulaciones usando los ángulos del robot y su correspondiente traducción quaternion por las traducciones derivadas. El variable alp sigue el parámetro $alfa$ en los parámetros DH y usamos esto en la determinación del eje de rotación. Esto puede tener valores 0 o cual $\frac{\pi}{2}$ hará cualquiera de los términos $-\sin(alp)$ y $\cos(alp)$ cero o ± 1 . Entonces, creamos el objeto de cuaternión dual combinando estos dos cuaterniones. Por último, almacenamos estos cuaterniones duales en una lista y la devolvemos a la anterior. método que realiza la transformación.

Cuaterniones duales de la matriz de transformación

Otro enfoque para derivar la cinemática directa puede ser mediante una transformación matriz. Esto se hace en el método derivar dual quat de TM. Lo normal La convención de derivar la cinemática directa es mediante el uso de matrices de rotación y vectores de traslación que se combinan para formar una matriz de transformación (4×4). A partir de esta matriz de transformación, podemos convertirla en un cuaternión dual convirtiendo las matrices de rotación (3×3) en cuaterniones unitarios y los vectores de traslación (3×1) en un cuaternión puro. La matriz de rotación a cuaternión la conversión se discutió en la Sección 3.4.2. Para la matriz de transformación, siga su derivación en [3] y [16].

```

1 # Obtiene la cuaternización dual para cada articulación
2 def deriveDualQuatFromTM(jangles):
3     print ( "Computing Transformation Matrix by Denavit-Hartenberg
4     Parámetros. . . " )
5
6 # Obtiene los ángulos de las articulaciones de la posición actual
7     theta = tintColor
8
9 # Obtiene matrices de transformación para cada 6 fotogramas

```

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

```

7 T = np. zeros ( forma = ( 6
                                , 4 , 4 ) )
8
9 # Construye las cuaterniones duales
10 DQLista = []
11
12 para mi in range ( 6 ) :
13     T[i] = ( [[ cos ( theta [ i ] ) , -sin ( theta [ i ] ) , a [ i ] ] [ sin ( theta [ i ] ) ,
14                 cos ( alpha a [ i ] ) , cos ( theta [ yo ] ) cos (
15     alfa a [ i ] ) , -sin ( alfa a [ i ] ) , -sin ( alfa a [ i ] ) d [ i ] ] [ sin ( theta
16                 [ i ] ) sin ( alfa a [ i ] ) , cos ( theta [ i ] ) pecado (
17     alfa a [ i ] ) , cos ( alfa a [ i ] ) , cos ( alfa a [ i ] ) d [ i ] ] ,
18                 [ 0 , 0 , 0 , 1 ] ] )
19
20     print ( "\n Transformación a Matrix for Frame" , i , " a " , yo + 1 , "\n"
21             , T[yo] )
22
23     # Deriva el Cuaternión Dual de la Transformación
24     Matriz
25     DualQuat = DualQuaternion ( Quaternion ( matriz = T[ i ] ) ,
26     Cuaternión ( vector = T[ i ] [ 0 : 3
27                 , 3 ] ) )
28     print ( "Cuaternión dual: ( [ "
29             , Dual Quat . q , " ] , , DualQuat . pag , ">"
30             " )
31
32     # Almacenar el Dual Quaternion en la lista
33     DQList. agregar (DualQuat)
34
35 devolver DQList

```

Igual que en el enfoque anterior, almacenamos los cuaterniones duales en una lista y páselo al primer método que realiza la transformación. el doble final el cuaternión que representa la transformación final será diferente del otro enfoque. Esto se debe a que en la derivación por AA, estamos basando la cuaterniones duales solo en el marco base mientras que para el enfoque TM, cada los cuaterniones duales se derivan con respecto a sus propios marcos. en cualquiera de estos enfoques, las coordenadas TCP finales siempre serán las mismas.

Aplicación práctica

Para la aplicación, usamos un método record pose que toma la corriente pose del robot y lo graba en un archivo de texto. Los coeficientes de este dual el cuaternión derivado de los ángulos de las articulaciones del robot se almacena en un texto archivo. Se pueden grabar varias poses en el archivo de texto. Aparte de esto, el La cinemática directa también se utiliza para filtrar las combinaciones de ángulos derivadas en la cinemática inversa comprobando si sus transformaciones equivalen a la transformación deseada escrita en el archivo de texto.

4.4. Aplicación práctica de programación

cinemática inversa

Para la cinemática inversa, nuestro objetivo es determinar los ángulos de articulación que darnos nuestra pose deseada (rotación y traslación). Por lo tanto, ahora se nos da la transformación final y la usamos para derivar los valores de los seis ángulos de articulación correspondientes. Usamos el método de obtener cinemática inversa para realizar la derivación. Comenzamos definiendo las coordenadas en el Cuadro 6 (Muñeca 3) y las coordenadas correspondientes en el Cuadro 5 (Muñeca 2).

```

1 # Da salida a ángulos de articulación de cinemáticos inversos
2 def getinversekinematics (pose deseada):
3     print ( "\nDe si pose roja : pose deseada . q ,          pose deseada. pag ) ;
4 # Transformación deseada del Cuadro 1 al 6
5 DQ16 = pose deseada
6 # Parte de rotación
7     w6 = DQ16. q [ 0 ]
8     a6 = DQ16. q [ 1 ]
9     b6 = DQ16. q [ 2 ]
10    c6 = DQ16. q [ 3 ]
11 # Transformación Part
12    x6 = DQ16. pag [ 1 ] ;
13    y6 = DQ16. pag [ 2 ] ;
14    z6 = DQ16. pag [ 3 ] ;
15
16 # Eliminar Transformación from Frame 5
17    DQ5i = Cuaternión Dual ( p=Cuaternión ( vector =[0 , d [ 5 ] , 0 ] ) )
18    DQ15p = DQ16. múltiple (DQ5i)
19    x5 = DQ15p. pag [ 1 ]
20    y5 = DQ15p. pag [ 2 ]
21    z5 = DQ15p. pag [ 3 ]

```

La traducción de la Muñeca 3 a la Muñeca 2 se demostró en la Sección 4.3.3.

Cálculo para Base (θ_1)

Para calcular θ_1 , usamos la ecuación (4.11):

$$\theta_1 = \text{atan2}(-x_5, y_5) \pm \text{atan2} \left(x_5^2 + y_5^2 - re_4^2, -d_4 \right)$$

Esto se expresa mediante el siguiente código,

```

1 # Calcular valores posibles para Theta 1
2     theta 1 =[]
3     t 1 1 = atan2 ( -x5 , y5 )+atan2 ( raíz cuadrada ( x5^2+y5^2-d [ 3 ]^2 ) ,-d [ 3 ] )
4     t 1 2 = atan2 ( -x5 , y5 )-atan2 ( raíz cuadrada ( x5^2+y5^2-d [ 3 ]^2 ) ,-d [ 3 ] )
5 # (base) Theta 1: 2 valores
6     teta 1 . agregar ( t 1 1 )
7     teta 1 . agregar ( t 1 2 )

```

Esto da como máximo dos valores posibles para θ_1 .

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

Cálculo para muñeca 2 (θ_5)

Para calcular θ_5 , usamos la ecuación (4.13):

$$\theta_5 = \pm \cos^{-1} \frac{x_6 \sin \theta_1 - y_6 \cos \theta_1 - d_4}{d_6}$$

Esto se expresa mediante el siguiente código,

```

1 # Calcular valores posibles para Theta 5
2 theta 1 5 = [ ]
3 para t 1 en theta 1 : # Para cada valor de Theta 1
4     # Evalúa te expres si onfor Theta 5
5     exp5 = ( x6 sen ( t 1 ) - y6 cos ( t 1 ) - d [ 3 ] ) / d [ 5 ]
6     # Comprueba si está dentro del dominio for arccos
7     si ( exp5 <= 1 y exp5 >= -1 ):
8         t 5 = acos ( exp5 )
9     elif (redondear (exp5) <= 1 y redondear (exp5) >= -1):
10        t 5 = acos (redondo (exp5))
11    otra cosa : conti nuo
12    # (Muñeca 2) Theta 5: 2 Valores
13    theta 1 5 . añadir ( [ t 1 t 5 ] )
14    theta 1 5 . añadir ( [ t 1 , -t 5 ] )

```

Consideramos que arccos tiene dominio de solo $[-1, 1]$. Por lo tanto, para prevenir errores de tiempo de ejecución, redondeamos los valores y comprobamos si ahora están dentro de este dominio. Esto da de nuevo dos valores posibles para θ_5 .

Cálculo para muñeca 3 (θ_6)

Para calcular θ_6 , usamos la ecuación (4.28):

$$\theta_6 = 2 \arctan 2(\gamma - \beta, \alpha + \delta)$$

dónde,

$$\alpha = \frac{wc^{-1} + cs^{-1}}{c^{-5}} \quad \beta = \frac{ac^{-1} + bs^{-1}}{s^{-5}} \quad \gamma = \frac{co^{-1} - bc^{-1}}{c^{-5}} \quad \delta = \frac{ws^{-1} - cc^{-1}}{s^{-5}}$$

Esto se expresa mediante el siguiente código,

```

1 # Calcular valores posibles para Theta 6
2 theta 1 5 6 = [ ]
3 para t 1 t 5 en theta 1 5 :
4     t 1 = t 1 t 5 [ 0 ]
5     t 5 = t 1 t 5 [ 1 ]
6
7     # Evita la singularidad para Theta 6
8     si (t 5 == 0) :

```

4.4. Aplicación práctica de programación

```

9      t156 = [ t1 si , t5 , 0 ]
10     t156 no está en theta 156 :
11         theta 156 . agregar ( t156 )
12     continuar
13
14     # Evalúa te expres si onfor Theta 6
15     tm1 = ( ( w6 cos ( t1 / 2 ) + c6 sin ( t1 / 2 ) ) / cos ( t5 / 2 ) )
16     tm2 = ( ( a6 cos ( t1 / 2 ) + b6 sen ( t1 / 2 ) ) / sen ( t5 / 2 ) )
17     tm3 = ( ( a6 sen ( t1 / 2 ) - b6 cos ( t1 / 2 ) ) / cos ( t5 / 2 ) )
18     tm4 = ( ( w6 sen ( t1 / 2 ) - c6 cos ( t1 / 2 ) ) / sen ( t5 / 2 ) )
19
20     # (Muñeca 3) Theta 6: 1 Valor
21     t6 = 2 atan2 ( tm3 - tm2 tm1 + tm4 )
22     theta 156 . añadir ( [ t1 t6 ] ) , t5 ,

```

Esto da solo un valor posible para θ_6 .

Ahora, para definir los siguientes ángulos. Necesitamos realizar una traducción de nuevo. Necesitamos obtener la transformación del Cuadro 2 (Hombro) al Cuadro 4 (Muñeca 1), por lo tanto necesitamos Q_2 . Esto se discutió en la Sección 4.3.4.

```

1 # Transformación derivada del cuadro 2 al cuadro 4
2     DQ24f = [ ]
3     para t156 en theta 156 :
4         t1 = t156 [ 0 ]
5         t5 = t156 [ 1 ]
6         t6 = t156 [ 2 ]
7         # Deriva el cuaternión dual inverso de las articulaciones 1 6 , 5 ,
8         q1 = np. concatenar ( ( [ cos ( t1 / 2 ) ] , np . multi p y ( sin ( t1 / 2 ) ,
9         [ 0 , 0 , -1 ] ) ) )
10        DQ1i = DualQuaternion ( Quaternion ( q1 ) , Quaternion ( vector
11        =[ 0 , 0 , -d [ 0 ] ] ) )
12        q5 = np. concatenar ( ( [ cos ( t5 / 2 ) ] , np . multi p y ( sin ( t5 / 2 ) ,
13        [ 0 , 0 , 1 ] ) ) )
14        DQ5i = Cuaternión dual ( Cuaternión ( q5 ) , Cuaternión ( vector =[ 0 ,
15        re , 0 ] ) )
16        [ 5 ] q6 = np . concatenar ( ( [ cos ( t6 / 2 ) ] , np . multi p y ( sin ( t6 / 2 ) ,
17        [ 0 , 1 , 0 ] ) ) )
18        DQ6i = Cuaternión dual ( Cuaternión ( q6 ) , Cuaternión ( vector
19        =[ 0 , 0 , 0 ] ) )
20        # Deriva la Transformación del Marco 2 al Marco 6
21        DQ26 = DQ1i . múltiple ( DQ16 )
22        # Deriva la Transformación Del Marco 2 al Marco 5
23        DQ25 = DQ26 . múltiple ( DQ6i )
24        # Deriva la Transformación del Marco 2 al Marco 4
25        DQ24 = DQ25 . múltiple ( DQ5i )
26        # Eliminar Transformación del Cuadro 4
27        DQ4i = Cuaternión Dual ( p=Cuaternión ( vector =[ 0 DQ24f . append , 0 , re [ 4 ] ] ) )
28        ( [ DQ24 . mult ( DQ4i ) , t156 ] )

```

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

Para cada combinación de ángulos de θ_1 , θ_5 y θ_6 , derivamos su correspondiente cuaternión dual inverso Q^{-1}_1 , Q^{-1}_5 y Q^{-1}_6 . Usamos estos inversos para derivar la transformación Q_2 a Q_4 . Entonces, revertimos la traducción hecha por Q_4 por también usando su inverso Q^{-1}_4 . Luego, las transformaciones se almacenan en una lista a lo largo de con el conjunto de ángulos $[\theta_1, \theta_5, \theta_6]$ que los generó. Todos estos serán utilizado en el cálculo de los ángulos de unión restantes.

Cálculo para codo (θ_3)

Para calcular θ_3 , usamos la ecuación (4.38):

$$\theta_3 = \pm \arccos \frac{x_4^2 + z_4^2 - u_3^2 - u_4^2}{2a_2a_3}$$

Esto se expresa mediante el siguiente código,

```

1 # Calcular valores posibles para Theta 3
2   el ta 1 3 5 6 q =[]
3   para dq en DQ24f:
4       w4 = dq [ 0 ] . q [ 0 ]
5       b4 = dq [ 0 ] . q [ 2 ]
6       x4 = dq [ 0 ] . pag [ 1 ]
7       y4 = dq [ 0 ] . pag [ 2 ]
8       z4 = dq [ 0 ] . pag [ 3 ]
9       q4 = [ w4 b4 , x4, y4, z4]
10
11      t1 = dq [ 1 ] [ 0 ]
12      t5 = dq [ 1 ] [ 1 ]
13      t6 = dq [ 1 ] [ 2 ]
14
15      # Evalua te expres si onfor Theta 3
16      exp3 = ( ( x4^2 + z4^2 ) - a[3]^2 - a[2]^2 ) / ( 2 * a[2] * a[3] )
17
18      # Comprueba si está dentro del dominio forarccos
19      si ( exp3 <= 1 y exp3 >= -1 ):
20          t3 = acos ( exp3 )
21      elif (redondear (exp3) <= 1 y redondear (exp3) >= -1):
22          t3 = acos (redondo ( exp3 ) )
23      otra cosa : conti nuo
24
25      # ( Codo ) Theta 3 : 2 Valores
26      el ta 1 3 5 6 q . agregar ( [ t1 t5 the t3 , q4 ] )
27      5 6 q . agregar ( [ t1 , -t3 , t5 t6 , q4 ] )

```

Lo mismo con θ_5 , también consideramos que los resultados están dentro del dominio de arccos. Esto da como máximo dos valores posibles para θ_3 .

4.4. Aplicación práctica de programación

Cálculo para hombro (θ_2)

Para calcular θ_2 , usamos la ecuación (4.41):

$$\theta_2 = \text{atan2}(z_4(a_2 + a_3 \cos \theta_3) - x_4 a_3 \sin \theta_3, x_4(a_2 + a_3 \cos \theta_3) + z_4 a_3 \sin \theta_3)$$

Esto se expresa mediante el siguiente código,

```

1 # Calcular valores posibles para Theta 2
2   el ta 1 2 3 5 6 q =[]
3   para t1356q en el ta 1 3 5 6 q :
4       t 1 = t1356q [ 0 ]
5       t 3 = t1356q [ 1 ]
6       t 5 = t1356q [ 2 ]
7       t 6 = t1356q [ 3 ]
8       x4 = t1356q [ 4 ] [ 2 ]
9       z4 = t1356q [ 4 ] [ 4 ]
10      # Ev alu a te expres si onfor Theta 2
11      t 2 = atan2 ( ( z4 ( un [ 2 ] + un [ 3 ] cos ( t 3 ) ) - x4 a [ 3 ] sin ( t 3 ) ) , ( x4 ( un
[ 2 ] + a [ 3 ] cos ( t 3 ) ) + z4 a [ 3 ] sin ( t 3 ) ) )
12      # ( Hombro ) Theta 2 : 1 Valor
13      el ta 1 2 3 5 6 q . añadir ( [ t1 t5 , t2 , t3 , , t6 , t1356q [ 4 ] ] )

```

Esto da solo un valor posible para θ_2 .

Cálculo para muñeca 1 (θ_4)

Para calcular θ_4 , usamos la ecuación (4.44):

$$\theta_4 = -2 \text{atan2}(w_4 s^-(2+3) + b_4 c^-(2+3), w_4 c^-(2+3) - b_4 s^-(2+3))$$

Esto se expresa mediante el siguiente código,

```

1 # Calcular valores posibles para Theta 4
2   el ta 1 2 3 4 5 6 =[]
3   para t12356q en el ta 1 2 3 5 6 q :
4       t 1 = t12356q [ 0 ]
5       t 2 = t12356q [ 1 ]
6       t 3 = t12356q [ 2 ]
7       t 5 = t12356q [ 3 ]
8       t 6 = t12356q [ 4 ]
9       w4 = t12356q [ 5 ] [ 0 ]
10      b4 = t12356q [ 5 ] [ 1 ]
11      # Ev alu a te expres si onfor Theta 4
12      t 4 = -2 atan2 ( ( w4 sen ( ( t 2 + t 3 ) / 2 ) ) + ( b4 cos ( ( t 2 + t 3 ) / 2 ) ) ) , ( w4
cos ( ( t 2 + t 3 ) / 2 ) ) - ( b4 sin ( ( t 2 + t 3 ) / 2 ) ) )
13      # ( Muñeca 1 ) Theta 4 : 1 Valor
14      el ta 1 2 3 4 5 6 . añadir ( [ t1 t3 , t2 , , t4 , t5 , t 6 ] )

```

Esto da solo un valor posible para θ_4 .

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

Filtración de combinaciones de ángulos y aplicación práctica

Para filtrar y verificar las combinaciones de ángulos resultantes, derivamos su cinemática directa correspondiente que representa la transformación final realizada por estos ángulos. Si esta transformación no es aproximadamente igual a nuestra transformación deseada, luego filtramos estos ángulos y dejamos solo aquellos que satisfacen nuestra igualdad dentro de la tolerancia de $1e-3$. Esto se hace a continuación.

```

1 # Obtener el cuaternión dual final correspondiente basado en el
  combinación de ángulos
2 tinteos = []
3 print ( "\nConjunto de posibles ángulos de unión: \n" para , el ta 1 2 3 4 5 6 )
4 theta en el ta 1 2 3 4 5 6 :
5     FinalDQ = getforward kinematic ( "AA" , theta)
6
7     print ( "\nComparación de cuaterniones duales: " )
8     imprimir ( " Postura deseada: \n ( [ " , DQ16. q , " ] , , DQ16. p , ">" )
9     print ( " Pose actual: \n ( [ " , FinalDQ. q , " ] , , FinalDQ. p , ">" )
10
11     si (DQ16. es igual a (FinalDQ)) :
12         imprimir ( "¡Criterios PASADOS!" )
13         tinteos agregar (theta)
14     else : imprimir ( "¡Criterios FALLIDOS!" )
15
16 print ( "\nÁngulos de unión final : " , tinteos )
17
18 tinteos de retorno

```

Para la aplicación, primero construimos el cuaternión dual representado por el coeficientes del archivo de texto generado por el método de pose de registro. Entonces, aplicamos la cinemática inversa y usamos los primeros ángulos de unión resultantes como entrada para el movimiento del robot. (También se puede hacer un criterio adicional para encontrar la combinación óptima de ángulos de articulación)

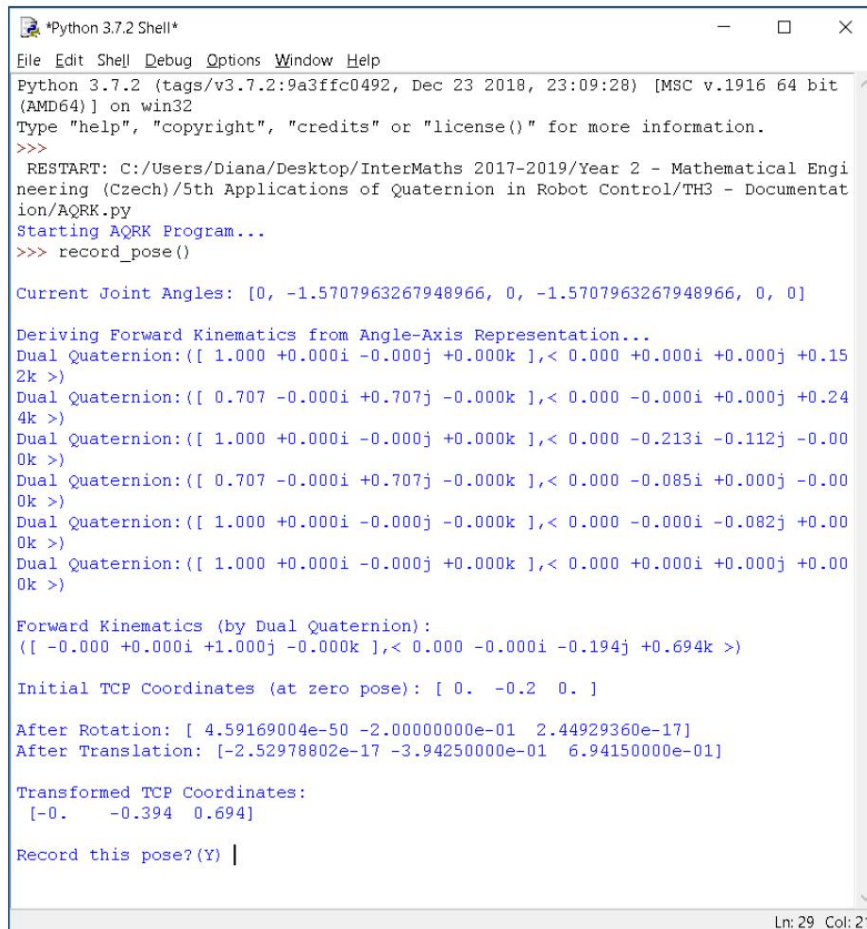
```

1 # Mueve al robot de la Transformación Dual Quaternion dada
2 def mover robot (dqcoeff): -
3     # Toma los coeficientes du al quater ni y crea el
4     du al quater ni on
5     DQ de si redpose = DualQuaternion ( Quaternion ( array=dqcoeff
6     [ 0 : 4 ] ) , Cuaternión ( array=dqcoeff [ 4 : 8 ] ) )
7
8     # A plica cinem á tico inverso para derivar los ángulos de articulación
9     ángulosarticulados = obtenercinemática inversa ( DQ de si redpose )
10
11     imprimir ( "\nHay " , len (ángulos de las articulaciones) , " posposibles!" )
12
13     );
14
15     # Mueve el ro b ot dados los ángulos de las articulaciones
16     robar movej (ángulos de articulación [ 0 ] acc =1, v el =1, w ai t=F al se )

```

4.5 Verificación de resultados

El programa comienza con la función de registro de pose:



```
*Python 3.7.2 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.2 (tags/v3.7.2:9a3ffc0492, Dec 23 2018, 23:09:28) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
RESTART: C:/Users/Diana/Desktop/InterMaths 2017-2019/Year 2 - Mathematical Engi
neering (Czech)/5th Applications of Quaternion in Robot Control/TH3 - Documentat
ion/AQRK.py
Starting AQRK Program...
>>> record_pose()

Current Joint Angles: [0, -1.5707963267948966, 0, -1.5707963267948966, 0, 0]

Deriving Forward Kinematics from Angle-Axis Representation...
Dual Quaternion:([ 1.000 +0.000i -0.000j +0.000k ],< 0.000 +0.000i +0.000j +0.15
2k >)
Dual Quaternion:([ 0.707 -0.000i +0.707j -0.000k ],< 0.000 -0.000i +0.000j +0.24
4k >)
Dual Quaternion:([ 1.000 +0.000i -0.000j +0.000k ],< 0.000 -0.213i -0.112j -0.00
0k >)
Dual Quaternion:([ 0.707 -0.000i +0.707j -0.000k ],< 0.000 -0.085i +0.000j -0.00
0k >)
Dual Quaternion:([ 1.000 +0.000i -0.000j -0.000k ],< 0.000 -0.000i -0.082j +0.00
0k >)
Dual Quaternion:([ 1.000 +0.000i -0.000j +0.000k ],< 0.000 +0.000i +0.000j +0.00
0k >)

Forward Kinematics (by Dual Quaternion):
([ -0.000 +0.000i +1.000j -0.000k ],< 0.000 -0.000i -0.194j +0.694k >)

Initial TCP Coordinates (at zero pose): [ 0. -0.2 0. ]

After Rotation: [ 4.59169004e-50 -2.00000000e-01 2.44929360e-17]
After Translation: [-2.52978802e-17 -3.94250000e-01 6.94150000e-01]

Transformed TCP Coordinates:
[-0.394 -0.394 0.694]

Record this pose?(Y) |
```

Esta es la pose base del robot dada por el siguiente conjunto de ángulos:

$$\begin{aligned}
 \text{Base: } \theta_1 &= 0 & \text{Hombro: } \theta_2 &= \frac{-\pi}{2} & \text{Codo: } \theta_3 &= 0 \\
 \text{Muñeca 1: } \theta_4 &= \frac{-\pi}{2} & \text{Muñeca 2: } \theta_5 &= 0 & \text{Muñeca 3: } \theta_6 &= 0
 \end{aligned}$$

Los cuaterniones duales anteriores se derivan con respecto a estos ángulos junto con su eje de rotación y sus correspondientes traslaciones. Esto se puede verificar en la Sección 4.2.1. Estos luego se multiplican y el producto se establece como el cuaternión dual que representa la transformación final. El punto central de la herramienta del efector final se transforma luego de las coordenadas (0, -0.2, 0) a (0, -0.394, 0.694) con respecto al marco base.

Capítulo 4. Aplicación de los cuaterniones a Universal Robot UR3

La pose de registro de función está configurada para usar la representación de eje-ángulo. Para fines de demostración, la derivación de la cinemática directa a partir de una matriz de transformación también se puede realizar llamando a la función de cinemática directa estableciendo los ángulos de unión como parámetros y TM como el modo de transformación. La implementación se demuestra a continuación:

```
>>> get_forward_kinematics("TM",rob.getj())

Current Joint Angles: [0, -1.5707963267948966, 0, -1.5707963267948966, 0, 0]

Deriving Forward Kinematics from Transformation Matrix...
Computing Transformation Matrix by Denavit-Hartenberg Parameters...

Transformation Matrix for Frame 0 to 1
[[ 1.    -0.    0.    0.   ]
 [ 0.    1.    -0.   -0.   ]
 [ 0.    0.    1.    0.1519]
 [ 0.    0.    0.    1.   ]]
Dual Quaternion:([ 1.000 +0.000i +0.000j +0.000k ],< 0.000 +0.000i -0.000j +0.152k >)

Transformation Matrix for Frame 1 to 2
[[ 6.12323400e-17  1.00000000e+00  0.00000000e+00  0.00000000e+00]
 [-6.12323400e-17  3.74939946e-33 -1.00000000e+00 -0.00000000e+00]
 [-1.00000000e+00  6.12323400e-17  6.12323400e-17  0.00000000e+00]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Dual Quaternion:([ 0.500 +0.500i +0.500j -0.500k ],< 0.000 +0.000i -0.000j +0.000k >)

Transformation Matrix for Frame 2 to 3
[[ 1.    -0.    0.   -0.24365]
 [ 0.    1.   -0.   -0.   ]
 [ 0.    0.    1.    0.   ]
 [ 0.    0.    0.    1.   ]]
Dual Quaternion:([ 1.000 +0.000i +0.000j +0.000k ],< 0.000 -0.244i -0.000j +0.000k >)

Transformation Matrix for Frame 3 to 4
[[ 6.123234e-17  1.000000e+00  0.000000e+00 -2.132500e-01]
 [-1.000000e+00  6.123234e-17 -0.000000e+00 -0.000000e+00]
 [ 0.000000e+00  0.000000e+00  1.000000e+00  1.123500e-01]
 [ 0.000000e+00  0.000000e+00  0.000000e+00  1.000000e+00]]
Dual Quaternion:([ 0.707 +0.000i +0.000j -0.707k ],< 0.000 -0.213i -0.000j +0.112k >)

Transformation Matrix for Frame 4 to 5
[[ 1.00000000e+00 -0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  6.12323400e-17 -1.00000000e+00 -8.53500000e-02]
 [ 0.00000000e+00  1.00000000e+00  6.12323400e-17  5.22618022e-18]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Dual Quaternion:([ 0.707 +0.707i +0.000j +0.000k ],< 0.000 +0.000i -0.085j +0.000k >)

Transformation Matrix for Frame 5 to 6
[[ 1.00000000e+00 -0.00000000e+00  0.00000000e+00  0.00000000e+00]
 [ 0.00000000e+00  6.12323400e-17  1.00000000e+00  8.19000000e-02]
 [-0.00000000e+00 -1.00000000e+00  6.12323400e-17  5.01492864e-18]
 [ 0.00000000e+00  0.00000000e+00  0.00000000e+00  1.00000000e+00]]
Dual Quaternion:([ 0.707 -0.707i +0.000j +0.000k ],< 0.000 +0.000i +0.082j +0.000k >)

Forward Kinematics (by Dual Quaternion):
([ 0.000 +0.000i +0.707j -0.707k ],< 0.000 +0.000i -0.194j +0.694k >)

Initial TCP Coordinates (at zero pose): [0. 0. 0.2]

After Rotation: [ 0. -0.2  0. ]
After Translation: [ 0. -0.39425  0.69415]

Transformed TCP Coordinates:
[ 0. -0.394  0.694]
```

El cuaternión dual que representa la transformación final en ambos enfoques es claramente diferente, pero las coordenadas transformadas son las mismas.

Tenga en cuenta que también estamos usando diferentes coordenadas TCP iniciales.

4.5. Verificación de Resultados

La función record pose almacena las poses en un archivo y la función run poses recupera los coeficientes de los cuaterniones duales almacenados en el archivo y genera los ángulos correspondientes a estas transformaciones. Esto se demuestra a continuación:

```
Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
>>> run_poses()

Desired Pose: -0.000 +0.000i +1.000j -0.000k 0.000 -0.000i -0.194j +0.694k

Set of Possible Joint Angles:
[[6.283185307179586, -1.570796340704604, 2.9802322387695312e-08, 4.7123889644920745,
0.0, 0], [6.283185307179586, -1.5707963128851894, -2.9802322387695312e-08, 4.712388996
277305, 0.0, 0], [0.0, -1.5707963407046037, 2.9802322387695312e-08, -1.570796342687512
, 0.0, 0], [0.0, -1.5707963128851894, -2.9802322387695312e-08, -1.5707963109022816, 0.
0, 0]]

Current Joint Angles: [6.283185307179586, -1.570796340704604, 2.9802322387695312e-08,
4.7123889644920745, 0.0, 0]

Deriving Forward Kinematics from Angle-Axis Representation...
Dual Quaternion: ([ -1.000 +0.000i -0.000j +0.000k ], < 0.000 +0.000i +0.000j +0.152k >)
Dual Quaternion: ([ 0.707 -0.000i +0.707j -0.000k ], < 0.000 +0.000i +0.000j +0.244k >)
Dual Quaternion: ([ 1.000 +0.000i -0.000j +0.000k ], < 0.000 -0.213i -0.112j -0.000k >)
Dual Quaternion: ([ -0.707 +0.000i -0.707j +0.000k ], < 0.000 -0.085i +0.000j +0.000k >)
Dual Quaternion: ([ 1.000 +0.000i -0.000j -0.000k ], < 0.000 -0.000i -0.082j +0.000k >)
Dual Quaternion: ([ 1.000 +0.000i -0.000j +0.000k ], < 0.000 +0.000i +0.000j +0.000k >)

Forward Kinematics (by Dual Quaternion):
([ -0.000 +0.000i +1.000j -0.000k ], < 0.000 +0.000i -0.194j +0.694k >)

Initial TCP Coordinates (at zero pose): [ 0. -0.2 0. ]

After Rotation: [-4.8985872e-17 -2.0000000e-01 2.4492936e-17]
After Translation: [-3.60044906e-17 -3.94250000e-01 6.94150000e-01]

Transformed TCP Coordinates:
[-0. -0.394 0.694]

Comparing Dual Quaternions:
Desired Pose:
([ -0.000 +0.000i +1.000j -0.000k ], < 0.000 -0.000i -0.194j +0.694k >)
Current Pose:
([ -0.000 +0.000i +1.000j -0.000k ], < 0.000 +0.000i -0.194j +0.694k >)
PASSED criteria!
```

Había 4 posibles combinaciones de ángulos. Esto se verifica aún más comprobando si su transformación correspondiente (representada por el cuaternión dual) es aproximadamente igual a nuestra transformación deseada como también se ve arriba.

Los resultados finales al final están dados por,

```
Final Joint Angles: [[6.283185307179586, -1.570796340704604, 2.9802322387695312e-08, 4.
7123889644920745, 0.0, 0], [6.283185307179586, -1.5707963128851894, -2.98023223876953
12e-08, 4.712388996277305, 0.0, 0], [0.0, -1.5707963407046037, 2.9802322387695312e-08,
-1.570796342687512, 0.0, 0], [0.0, -1.5707963128851894, -2.9802322387695312e-08, -1.57
07963109022816, 0.0, 0]]

There are 4 possible pose(s)!
>>>
```

Por lo tanto, todas las combinaciones fueron válidas y todas ellas transformarán el punto central de la herramienta de nuestro robot a la pose deseada (mismas coordenadas TCP transformadas) que es la solución de la cinemática inversa.

Capítulo 5

Conclusiones y Recomendaciones

5.1 Conclusión

La tesis pudo demostrar con éxito el uso del álgebra de cuaterniones en la representación de cinemáticas para aplicaciones robóticas en particular con el robot universal UR3 de 6 grados de libertad. Tanto su cinemática directa como inversa se derivaron utilizando los principios de los cuaterniones. Estos están contenidos en Sección 4.2 Cinemática Directa por Cuaterniones Duales y Sección 4.3 Inversa Cinemática por Cuaterniones Duales. Una aplicación práctica de programación fue también se desarrolló como se muestra en la Sección 4.4.2 Aplicación de programación Python para confirmar la precisión de la cinemática derivada.

Aparte de esto, en base a nuestro análisis de los resultados, también podemos concluir las siguientes declaraciones:

1. Los cuaterniones duales presentan una forma más compacta de representar transformaciones tridimensionales. Por lo tanto, también ofrece una ventaja considerable en términos de eficiencia de almacenamiento.
 - (a) Esto se debe a que los cuaterniones requieren solo cuatro elementos para representar una rotación en comparación con la rotación más utilizada. matrices que involucra 9 elementos.
 - (b) Para la transformación en tres dimensiones que implica rotación y traslación. Una matriz de transformación homogénea consta de 16 elementos en los que cuatro elementos son triviales, mientras que una dual el cuaternión solo necesita 7 elementos para representar la misma transformación.

Capítulo 5. Conclusiones y recomendaciones

2. La realización de transformaciones múltiples por cuaterniones duales también tiene una ventaja en términos de solidez computacional y precisión numérica, especialmente al tratar con la cinemática de un robot.
 - (a) La combinación de transformaciones utilizando matrices de transformación homogéneas implica multiplicar matrices de 4×4 , lo que implica varias operaciones aritméticas y trigonométricas que pueden disminuir considerablemente la precisión numérica de la solución.
 - (b) Realizar la multiplicación de cuaterniones es comparable a la multiplicación de polinomios y al mismo tiempo incorporar las leyes de multiplicación para las unidades de cuaterniones. Esto lo hace más eficiente y también más preciso numéricamente ya que implica menos operaciones aritméticas y trigonométricas.
3. Los cuaterniones también ofrecen una gran ventaja en términos de normalización y facilidad de derivación.
 - (a) Los cuaterniones son más fáciles de normalizar que las matrices. Una matriz de rotación tiene que ser ortogonal y debe tener un determinante de 1. Para los cuaterniones, tiene que ser un cuaternión unitario, lo que significa que tiene que tener una norma unitaria. Esto se puede hacer simplemente dividiendo los cuatro elementos de los cuaterniones por su magnitud.
 - (b) Los cuaterniones duales se derivan fácilmente usando la representación eje-ángulo en la que solo necesitamos determinar la relación del eje de rotación actual con el marco de coordenadas base.
4. Los cuaterniones duales no son susceptibles al bloqueo de cardán.
 - (a) Aunque no se discute en este estudio, el bloqueo del cardán es un problema serio al tratar con rotaciones en el espacio tridimensional. Imagine tres cardanes que representan rotaciones sobre cada tres ejes; por lo tanto, representa los tres grados de libertad. Estos se definen como balanceo, cabeceo y guiñada. Ocurre cuando los cardanes externo e interno se alinean entre sí y pierden un grado de libertad. Esto se debe al hecho de que la rotación que puede realizar cualquiera de estos dos gimbals ahora se considera la misma. Euler Angles es famoso por ser susceptible a esta rara ocurrencia.
 - (b) Se sabe que los cuaterniones, al igual que las matrices de rotación, no son susceptibles al bloqueo cardánico. Esto se debe a que representan la orientación como un valor en lugar de tres valores separados y relacionados.

5. La facilidad para representar la cinemática del robot mediante cuaterniones duales depende del tipo de cinemática que se derive y de la estructura cinemática del robot.

- (a) Para la cinemática directa, es aconsejable derivar las rotaciones mediante un enfoque de eje-ángulo en comparación con partir de una matriz de rotación y luego convertirla en cuaterniones unitarios. De esta manera, claramente habrá menos operaciones a realizar en comparación con la transformación homogénea tradicional.
- (b) Para la cinemática inversa, la dificultad de derivar fórmulas para los ángulos aún se conserva en los cuaterniones duales. Esto generalmente depende de la estructura cinemática del robot, como en nuestro caso. Pertenece a la relación de la transformación que puede realizar cada articulación con la transformación final. En nuestro caso, tuvimos que dividir las transformaciones en tres. Sin embargo, en algunos casos, puede suceder que ciertas fórmulas ya puedan derivarse solo de las ecuaciones generadas en la transformación final.
- (c) Para seguir aplicando el concepto de cuaterniones en la cinemática inversa, también tuvimos que analizar y calcular varias ecuaciones que pueden representar las transformaciones realizadas por cada articulación. Esto lo hace más tedioso en comparación con la convención normal que generalmente se realiza utilizando el enfoque de solución geométrica.

Con suerte, estas ventajas se considerarán suficientes para que los cuaterniones duales se utilicen más ampliamente no solo en el campo de la robótica y los gráficos por computadora, sino también en otras áreas que implican trabajar en el espacio tridimensional.

5.2 Recomendación

Para futuras investigaciones, se recomienda encontrar si existe una forma mejor y más eficiente de representar la cinemática del robot, particularmente para la cinemática inversa. Para Universal Robot UR3, la solución de cinemática inversa se puede filtrar aún más considerando solo un cierto rango de movimientos.

También se pueden aplicar ciertas condiciones para determinar la solución óptima que dará la pose deseada. Como esta investigación se centró solo en la aplicación de cuaterniones en la cinemática de robots, puede extenderse aún más a su aplicación en el control de robots en general. Esto puede implicar un paso adicional de realizar la interpolación de cuaterniones. Esto contribuye a encontrar el camino óptimo para la trayectoria del robot. Esto será bueno para futuras investigaciones ya que también es una de las aplicaciones útiles y conocidas de los cuaterniones.

Apéndice

Apéndice A

Robot universal UR3

Especificaciones técnicas

UR3

Actuación

Repetibilidad	±0,1 mm / ±0,0039 pulg. (4 milésimas de pulgada)
Rango de temperatura ambiente	0-50° *
El consumo de energía	Min. 90 W, Típico 125 W, Máx. 250 W
Operación de colaboración	15 funciones de seguridad ajustables avanzadas. Función de seguridad aprobada por TÜV NORD Probado de acuerdo con: EN ISO 13849:2008 PLd

Especificación

Carga útil	3 kg / 6,6 libras
Alcanzar	500 mm/19,7 pulgadas
Grados de libertad	6 juntas giratorias
Programación	Interfaz gráfica de usuario Polyscope en pantalla táctil de 12 pulgadas con montaje

Movimienot

Brazo robótico de movimiento de ejes	Rango de trabajo	Velocidad máxima
Base	± 360°	± 180°/seg.
Hombro	± 360°	± 180°/seg.
Codo	± 360°	± 180°/seg.
Muñeca 1	± 360°	± 360°/seg.
Muñeca 2	± 360°	± 360°/seg.
Muñeca 3	Infinito	±360°/seg.
Herramienta típica		1 m/seg. / 39,4 pulg./seg.

Características

clasificación IP	IP64	
Sala limpia de clase ISO	5	
Ruido	70dB	
Montaje de robots	Cualquier	
puertos de E/S	entrada digital	2
	Salida digital	2
	entrada analógica	2
	salida analógica	0
Fuente de alimentación de E/S en la herramienta	12 V/24 V 600 mA en herramienta	

Físico

Huella	Ø128mm
Materiales	Aluminio, plásticos PP
Tipo de conector de herramienta	M8
Brazo robótico de longitud de cable	6 metros / 236 pulgadas
Peso con cable	11 kg /24,3 libras

* El robot puede trabajar en un rango de temperatura de 0-50°C. A alta velocidad continua de la junta, la temperatura ambiente se reduce.

CAJA DE CONTROL

Características

clasificación IP	IP20	
Sala limpia de clase ISO	6	
Ruido	<65dB(A)	
puertos de E/S	entrada digital	2
	Salida digital	2
	entrada analógica	2
	salida analógica	2
Fuente de alimentación de E/S	24V 2A	
Comunicación	TCP/IP 100 Mbit, Modbus TCP, Profinet, Ethernet IP	
Fuente de alimentación	100-240 VCA, 50-60 Hz	
Rango de temperatura ambiente	0-50°	
Físico		
Tamaño de la caja de control (An. x Al. x Pr.)	475 mm x 423 mm x 268 mm / 18,7 x 16,7 x 10,6 pulgadas	
Peso	15 kg / 33,1 libras	
Materiales	Acero	

COLGANTE ENSEÑANZA

Características

clasificación IP	IP20
Físico	
Materiales	Aluminio, PP
Peso	1,5 kg / 3,3 libras
Longitud del cable	4,5 m / 177 pulgadas



apéndice B

Productos de transformación para Base a Muñeca 3

Esta parte muestra los productos de transformación para la cinemática del robot UR3 estructura desde el Marco 1 (Base) hasta el Marco 6 (Muñeca 3).

Definimos el producto dual de cuaterniones M_i como:

$$M_i = Q_i \quad M_{i+1} \text{ donde } 1 \leq i \leq 6$$

$$M_6 = Q_6$$

$$M_5 = Q_5 \quad M_6 = Q_5 \quad Q_6$$

$$M_4 = Q_4 \quad M_5 = Q_4 \quad Q_5 \quad Q_5$$

$$M_3 = Q_3 \quad M_4 = Q_3 \quad Q_4 \quad Q_5 \quad Q_6$$

$$M_2 = Q_2 \quad M_3 = Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6$$

$$M_1 = Q_1 \quad M_2 = Q_1 \quad Q_2 \quad Q_3 \quad Q_4 \quad Q_5 \quad Q_6$$

Definimos el producto de cuaternión dual N_i de la siguiente manera:

$$N_{i+1} = Q^{-1}_i \quad N_i \text{ donde } 1 \leq i \leq 6$$

$$N_1 = QFK = ([w, < a, b, c >], < x, y, z >)$$

$$N_2 = Q^{-1}_1 \quad N_1 = Q^{-1}_1 \quad QFK$$

$$N_3 = Q^{-1}_2 \quad N_2 = Q^{-1}_2 \quad Q^{-1}_1 \quad QFK$$

$$N_4 = Q^{-1}_3 \quad N_3 = Q^{-1}_3 \quad Q^{-1}_2 \quad Q^{-1}_1 \quad QFK$$

$$N_5 = Q^{-1}_4 \quad N_4 = Q^{-1}_4 \quad Q^{-1}_3 \quad Q^{-1}_2 \quad Q^{-1}_1 \quad QFK$$

$$N_6 = Q^{-1}_5 \quad N_5 = Q^{-1}_5 \quad Q^{-1}_4 \quad Q^{-1}_3 \quad Q^{-1}_2 \quad Q^{-1}_1 \quad QFK$$

 Apéndice B. Productos de transformación para Base toWrist 3

 Productos de Transformación de Base a Muñeca 3 (M6 = N6)

$$M6 = ([M61 + M62i + M63j + M64k], < M65i + M66j + M67k >)$$

$$N6 = ([N61 + N62i + N63j + N64k], < N65i + N66j + N67k >)$$

$$\begin{array}{lcl}
 & \underline{M61 = N61} & \\
 \text{porque } \frac{\theta 6}{2} = w \text{ porque } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{porque } \frac{\theta 1 - \theta 5}{2} & (B.1) \\
 & \text{+un pecado } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{pecado } \frac{\theta 1 + \theta 5}{2} \\
 & \text{-b pecado } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{porque } \frac{\theta 1 + \theta 5}{2} \\
 & \text{+c porque } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{pecado } \frac{\theta 1 - \theta 5}{2}
 \end{array}$$

$$\begin{array}{lcl}
 & \underline{M62 = N62} & \\
 0 = -w \text{ sen } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{pecado } \frac{\theta 1 + \theta 5}{2} & (B.2) \\
 & \text{+a porque } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{porque } \frac{\theta 1 - \theta 5}{2} \\
 & \text{+b porque } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{pecado } \frac{\theta 1 - \theta 5}{2} \\
 & \text{+c pecado } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{porque } \frac{\theta 1 + \theta 5}{2}
 \end{array}$$

$$\begin{array}{lcl}
 & \underline{M63 = N63} & \\
 - \text{pecado } \frac{\theta 6}{2} = w \text{ pecado } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{porque } \frac{\theta 1 + \theta 5}{2} & (B.3) \\
 & \text{-a porque } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{pecado } \frac{\theta 1 - \theta 5}{2} \\
 & \text{+b porque } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{porque } \frac{\theta 1 - \theta 5}{2} \\
 & \text{+c pecado } \frac{\theta 2 + \theta 3 + \theta 4}{2} & \text{pecado } \frac{\theta 1 + \theta 5}{2}
 \end{array}$$

$$\underline{M64 = N64}$$

$$\begin{aligned}
 0 = -w \cos & \frac{\theta_2 + \theta_3 + \theta_4}{2} \quad \text{pecado} \quad \frac{\theta_1 - \theta_5}{2} & (B.4) \\
 -un \text{ pecado} & \frac{\theta_2 + \theta_3 + \theta_4}{2} \quad \text{porque} \quad \frac{\theta_1 + \theta_5}{2} \\
 -b \text{ pecado} & \frac{\theta_2 + \theta_3 + \theta_4}{2} \quad \text{pecado} \quad \frac{\theta_1 + \theta_5}{2} \\
 +c \text{ porque} & \frac{\theta_2 + \theta_3 + \theta_4}{2} \quad \text{porque} \quad \frac{\theta_1 - \theta_5}{2}
 \end{aligned}$$

$$\underline{M65 = N65}$$

$$\begin{aligned}
 0 = & -d_1 \cos \theta_5 \sin (\theta_2 + \theta_3 + \theta_4) - d_4 \sin \theta_5 - a_2 \cos (\theta_3 + \theta_4) \cos \theta_5 \quad (B.5) \\
 & - a_3 \cos \theta_4 \cos \theta_5 + x [\cos \theta_1 \cos (\theta_2 + \theta_3 + \theta_4) \cos \theta_5 + \sin \theta_1 \sin \theta_5] \\
 & + y [\cos (\theta_2 + \theta_3 + \theta_4) \cos \theta_5 \sin \theta_1 - \cos \theta_1 \sin \theta_5] \\
 & + z \cos \theta_5 \sin (\theta_2 + \theta_3 + \theta_4)
 \end{aligned}$$

$$\underline{M66 = N66}$$

$$\begin{aligned}
 0 = & -d_1 \sin (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 + d_4 \cos \theta_5 - a_2 \cos (\theta_3 + \theta_4) \sin \theta_5 \quad (B.6) \\
 & - a_3 \cos \theta_4 \sin \theta_5 + x [\cos \theta_1 \cos (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 - \cos \theta_5 \sin \theta_1] \\
 & + y [\cos (\theta_2 + \theta_3 + \theta_4) \sin \theta_1 \sin \theta_5 + \cos \theta_1 \cos \theta_5] \\
 & + z \sin (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 + d_6
 \end{aligned}$$

$$\underline{M67 = N67}$$

$$\begin{aligned}
 0 = & -d_1 \cos (\theta_2 + \theta_3 + \theta_4) - x \cos \theta_1 \sin (\theta_2 + \theta_3 + \theta_4) - y \sin & (B.7) \\
 & \theta_1 \sin (\theta_2 + \theta_3 + \theta_4) + z \cos (\theta_2 + \theta_3 + \theta_4) \\
 & + a_2 \sin (\theta_3 + \theta_4) + a_3 \sin \theta_4 + d_5
 \end{aligned}$$

Productos de Transformación de Base a Muñeca 3 (M5 = N5)

$$M5 = ([M51 + M52i + M53j + M54k], < M55i + M56j + M57k >)$$

$$N5 = ([N51 + N52i + N53j + N54k], < N55i + N56j + N57k >)$$

$$\underline{M51 = N51}$$

$$\begin{aligned} \text{porque } \frac{\theta_5}{2} \quad \text{porque } \frac{\theta_6}{2} &= w \text{ porque } \frac{\theta_1}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4}{2} & (B.8) \\ &+ \text{un pecado } \frac{\theta_1}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4}{2} \\ &-b \text{ porque } \frac{\theta_1}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4}{2} \\ &+c \text{ pecado } \frac{\theta_1}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4}{2} \end{aligned}$$

$$\underline{M52 = N52}$$

$$\begin{aligned} - \text{pecado } \frac{\theta_5}{2} \quad \text{pecado } \frac{\theta_6}{2} &= -w \text{ sen } \frac{\theta_1}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4}{2} & (B.9) \\ &+a \text{ porque } \frac{\theta_1}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4}{2} \\ &+b \text{ pecado } \frac{\theta_1}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4}{2} \\ &+c \text{ porque } \frac{\theta_1}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4}{2} \end{aligned}$$

$$\underline{M53 = N53}$$

$$\begin{aligned} - \text{porque } \frac{\theta_5}{2} \quad \text{pecado } \frac{\theta_6}{2} &= w \text{ porque } \frac{\theta_1}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4}{2} & (B.10) \\ &- \text{un pecado } \frac{\theta_1}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4}{2} \\ &+b \text{ porque } \frac{\theta_1}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4}{2} \\ &+c \text{ pecado } \frac{\theta_1}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4}{2} \end{aligned}$$

$$M54 = N54$$

$$\begin{aligned}
 -\text{pecado} \frac{\theta_5}{2} \text{ porque } \frac{\theta_6}{2} &= -w \text{ sen } \frac{\theta_1}{2} \text{ porque } \frac{\theta_2 + \theta_3 + \theta_4}{2} & (B.11) \\
 -a \text{ porque } \frac{\theta_1}{2} &\text{ pecado } \frac{\theta_2 + \theta_3 + \theta_4}{2} \\
 -b \text{ pecado } \frac{\theta_1}{2} &\text{ pecado } \frac{\theta_2 + \theta_3 + \theta_4}{2} \\
 +c \text{ porque } \frac{\theta_1}{2} &\text{ porque } \frac{\theta_2 + \theta_3 + \theta_4}{2}
 \end{aligned}$$

$$M55 = N55$$

$$\begin{aligned}
 -d_6 \text{ sen } \theta_5 &= -d_1 \text{ sen } (\theta_2 + \theta_3 + \theta_4) - a_2 \cos (\theta_3 + \theta_4) - a_3 \cos \theta_4 & (B.12) \\
 &+ x \cos \theta_1 \cos (\theta_2 + \theta_3 + \theta_4) + y \text{ sen } \theta_1 \cos (\theta_2 + \theta_3 + \theta_4) \\
 &+ z \text{ sen } (\theta_2 + \theta_3 + \theta_4)
 \end{aligned}$$

$$M56 = N56$$

$$-d_6 \cos \theta_5 = y \cos \theta_1 - x \text{ sen } \theta_1 + d_4 \quad (B.13)$$

$$M57 = N57$$

$$\begin{aligned}
 0 &= -d_1 \cos (\theta_2 + \theta_3 + \theta_4) + d_5 + a_2 \text{ sen } (\theta_3 + \theta_4) + a_3 \text{ sen } \theta_4 - x \cos \theta_1 & (B.14) \\
 &\text{ sen } (\theta_2 + \theta_3 + \theta_4) - y \text{ sen } \theta_1 \text{ sen } (\theta_2 + \theta_3 + \theta_4) \\
 &+ z \text{ porque } (\theta_2 + \theta_3 + \theta_4)
 \end{aligned}$$

Productos de Transformación de Base a Muñeca 3 (M4 = N4)

$$M4 = ([M41 + M42i + M43j + M44k], < M45i + M46j + M47k >)$$

$$N4 = ([N41 + N42i + N43j + N44k], < N45i + N46j + N47k >)$$

$$\underline{M41 = N41}$$

$$\begin{aligned} \text{porque } \frac{\theta 5}{2} \quad \text{porque } \frac{\theta 4 + \theta 6}{2} &= w \text{ porque } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2 + \theta 3}{2} & (B.15) \\ + \text{un pecado } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2 + \theta 3}{2} \\ -b \text{ porque } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2 + \theta 3}{2} \\ +c \text{ pecado } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2 + \theta 3}{2} \end{aligned}$$

$$\underline{M42 = N42}$$

$$\begin{aligned} \text{pecado } \frac{\theta 5}{2} \quad \text{pecado } \frac{\theta 4 - \theta 6}{2} &= -w \text{ sen } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2 + \theta 3}{2} & (B.16) \\ +a \text{ porque } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2 + \theta 3}{2} \\ +b \text{ pecado } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2 + \theta 3}{2} \\ +c \text{ porque } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2 + \theta 3}{2} \end{aligned}$$

$$\underline{M43 = N43}$$

$$\begin{aligned} - \text{porque } \frac{\theta 5}{2} \quad \text{pecado } \frac{\theta 4 + \theta 6}{2} &= w \text{ porque } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2 + \theta 3}{2} & (B.17) \\ - \text{un pecado } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2 + \theta 3}{2} \\ +b \text{ porque } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2 + \theta 3}{2} \\ +c \text{ pecado } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2 + \theta 3}{2} \end{aligned}$$

$$\underline{M44 = N44}$$

$$\begin{aligned}
 -\text{pecado} \frac{\theta_5}{2} \text{ porque } \frac{\theta_4 - \theta_6}{2} &= -w \text{ sen } \frac{\theta_1}{2} \text{ porque } \frac{\theta_2 + \theta_3}{2} & (B.18) \\
 -a \text{ porque } \frac{\theta_1}{2} &\text{ pecado } \frac{\theta_2 + \theta_3}{2} \\
 -b \text{ pecado } \frac{\theta_1}{2} &\text{ pecado } \frac{\theta_2 + \theta_3}{2} \\
 +c \text{ porque } \frac{\theta_1}{2} &\text{ porque } \frac{\theta_2 + \theta_3}{2}
 \end{aligned}$$

$$\underline{M45 = N45}$$

$$\begin{aligned}
 -d_6 \cos \theta_4 \sin \theta_5 + d_5 \sin \theta_4 &= -d_1 \sin (\theta_2 + \theta_3) - a_2 \cos \theta_3 - a_3 + x \cos & (B.19) \\
 &\theta_1 \cos (\theta_2 + \theta_3) \\
 &+ y \sin \theta_1 \cos (\theta_2 + \theta_3) \\
 &+ z \sin (\theta_2 + \theta_3)
 \end{aligned}$$

$$\underline{M46 = N46}$$

$$-d_6 \cos \theta_5 = y \cos \theta_1 - x \sin \theta_1 + d_4 \quad (B.20)$$

$$\underline{M47 = N47}$$

$$\begin{aligned}
 -d_6 \sin \theta_4 \sin \theta_5 - d_5 \cos \theta_4 &= -d_1 \cos (\theta_2 + \theta_3) + a_2 \sin \theta_3 - x \cos & (B.21) \\
 &\theta_1 \sin (\theta_2 + \theta_3) \\
 &- y \sin \theta_1 \sin (\theta_2 + \theta_3) \\
 &+ z \text{ porque } (\theta_2 + \theta_3)
 \end{aligned}$$

 Apéndice B. Productos de transformación para Base toWrist 3

 Productos de Transformación de Base a Muñeca 3 ($M3 = N3$)

$$M3 = ([M31 + M32i + M33j + M34k], < M35i + M36j + M37k >)$$

$$N3 = ([N31 + N32i + N33j + N34k], < N35i + N36j + N37k >)$$

$$\underline{M31 = N31}$$

$$\begin{aligned} \text{porque } \frac{\theta 5}{2} \quad \text{porque } \frac{\theta 3 + \theta 4 + \theta 6}{2} &= w \text{ porque } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2}{2} & (B.22) \\ &+ \text{un pecado } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2}{2} \\ &-b \text{ porque } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2}{2} \\ &+c \text{ pecado } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2}{2} \end{aligned}$$

$$\underline{M32 = N32}$$

$$\begin{aligned} \text{pecado } \frac{\theta 5}{2} \quad \text{pecado } \frac{\theta 3 + \theta 4 - \theta 6}{2} &= -w \text{ sen } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2}{2} & (B.23) \\ &+a \text{ porque } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2}{2} \\ &+b \text{ pecado } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2}{2} \\ &+c \text{ porque } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2}{2} \end{aligned}$$

$$\underline{M33 = N33}$$

$$\begin{aligned} - \text{ porque } \frac{\theta 5}{2} \quad \text{pecado } \frac{\theta 3 + \theta 4 + \theta 6}{2} &= w \text{ porque } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2}{2} & (B.24) \\ &- \text{un pecado } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2}{2} \\ &+b \text{ porque } \frac{\theta 1}{2} \quad \text{porque } \frac{\theta 2}{2} \\ &+c \text{ pecado } \frac{\theta 1}{2} \quad \text{pecado } \frac{\theta 2}{2} \end{aligned}$$

$$\underline{M34 = N34}$$

$$\begin{aligned} -\text{pecado } \frac{\theta_5}{2} \text{ porque } \frac{\theta_3 + \theta_4 - \theta_6}{2} &= -w \text{ sen } \frac{\theta_1}{2} \text{ porque } \frac{\theta_2}{2} \\ -a \text{ porque } \frac{\theta_1}{2} \text{ pecado } \frac{\theta_2}{2} \\ -b \text{ pecado } \frac{\theta_1}{2} \text{ pecado } \frac{\theta_2}{2} \\ +c \text{ porque } \frac{\theta_1}{2} \text{ porque } \frac{\theta_2}{2} \end{aligned} \quad (B.25)$$

$$\underline{M35 = N35}$$

$$\begin{aligned} -d_6 \cos(\theta_3 + \theta_4) \text{ sen } \theta_5 + d_5 \text{ sen }(\theta_3 + \theta_4) + a_3 \cos \theta_3 &= -d_1 \\ \text{sen } \theta_2 - a_2 + x \cos \theta_1 \cos \theta_2 \\ + y \text{ sen } \theta_1 \cos \theta_2 + z \text{ sen } \theta_2 \end{aligned} \quad (B.26)$$

$$\underline{M36 = N36}$$

$$-d_6 \cos \theta_5 - d_4 = y \cos \theta_1 - x \text{ sen } \theta_1 \quad (B.27)$$

$$\underline{M37 = N37}$$

$$\begin{aligned} -d_6 \text{ sen }(\theta_3 + \theta_4) \text{ sen } \theta_5 - d_5 \cos(\theta_3 + \theta_4) + a_3 \text{ sen } \theta_3 &= \\ -d_1 \cos \theta_2 - x \cos \theta_1 \text{ sen } \theta_2 \\ -y \text{ sen } \theta_1 \text{ sen } \theta_2 + z \cos \theta_2 \end{aligned} \quad (B.28)$$

Apéndice B. Productos de transformación para Base toWrist 3 ---

Productos de Transformación de Base a Muñeca 3 (M2 = N2) ---

$$M2 = ([M21 + M22i + M23j + M24k], < M25i + M26j + M27k >)$$

$$N2 = ([N21 + N22i + N23j + N24k], < N25i + N26j + N27k >)$$

$$\underline{M21 = N21}$$

$$\text{porque } \frac{\theta_5}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} = w \text{ porque } \frac{\theta_1}{2} + c \text{ pecado } \frac{\theta_1}{2} \quad (B.29)$$

$$\underline{M22 = N22}$$

$$\text{pecado } \frac{\theta_5}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} = \text{un porque } \frac{\theta_1}{2} + b \text{ pecado } \frac{\theta_1}{2} \quad (B.30)$$

$$\underline{M23 = N23}$$

$$- \text{porque } \frac{\theta_5}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} = -\text{un pecado } \frac{\theta_1}{2} + b \text{ porque } \frac{\theta_1}{2} \quad (B.31)$$

$$\underline{M24 = N24}$$

$$- \text{pecado } \frac{\theta_5}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} = -w \text{ sen } \frac{\theta_1}{2} + c \text{ porque } \frac{\theta_1}{2} \quad (B.32)$$

$$\underline{M25 = N25}$$

$$-d_6 \cos(\theta_2 + \theta_3 + \theta_4) \sin \theta_5 + d_5 \sin(\theta_2 + \theta_3 + \theta_4) + a_3 \quad (B.33)$$

$$\cos(\theta_2 + \theta_3) + a_2 \cos \theta_2 = x \cos \theta_1 + y \sin \theta_1$$

$$\underline{M26 = N26}$$

$$-d_6 \cos \theta_5 - d_4 = y \cos \theta_1 - x \sin \theta_1 \quad (B.34)$$

$$\underline{M27 = N27}$$

$$-d_6 \sin(\theta_2 + \theta_3 + \theta_4) \sin \theta_5 - d_5 \cos(\theta_2 + \theta_3 + \theta_4) + a_3 \quad (B.35)$$

$$\sin(\theta_2 + \theta_3) + a_2 \sin \theta_2 = -d_1 + z$$

Productos de Transformación de Base a Muñeca 3 (M1 = N1)

$$M1 = ([M11 + M12i + M13j + M14k], < M15i + M16j + M17k >)$$

$$N1 = ([N11 + N12i + N13j + N14k], < N15i + N16j + N17k >)$$

$$\underline{M11 = N11}$$

$$\text{porque } \frac{\theta_1 - \theta_5}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} = w \quad (B.36)$$

$$\underline{M12 = N12}$$

$$\text{pecado } \frac{\theta_1 + \theta_5}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} = un \quad (B.37)$$

$$\underline{M13 = N13}$$

$$- \text{porque } \frac{\theta_1 + \theta_5}{2} \quad \text{pecado } \frac{\theta_2 + \theta_3 + \theta_4 + \theta_6}{2} = \text{segundo} \quad (B.38)$$

$$\underline{M14 = N14}$$

$$- \text{pecado } \frac{\theta_5 - \theta_1}{2} \quad \text{porque } \frac{\theta_2 + \theta_3 + \theta_4 - \theta_6}{2} = do \quad (B.39)$$

$$\underline{M15 = N15}$$

$$\begin{aligned} -d_6 \cos \theta_1 \cos (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 + d_6 \cos \theta_5 \sin \theta_1 + d_4 \sin \theta_1 + d_5 \\ \cos \theta_1 \sin (\theta_2 + \theta_3 + \theta_4) + a_3 \cos \theta_1 \cos (\theta_2 + \theta_3) \\ + a_2 \cos \theta_1 \cos \theta_2 = x \end{aligned} \quad (B.40)$$

$$\underline{M16 = N16}$$

$$\begin{aligned} -d_6 \cos \theta_1 \cos \theta_5 - d_4 \cos \theta_1 + d_6 \cos (\theta_2 + \theta_3 + \theta_4) \sin \theta_1 \sin \theta_5 + d_5 \\ \sin (\theta_2 + \theta_3 + \theta_4) + a_3 \cos (\theta_2 + \theta_3) + a_2 \cos \theta_2 = y \end{aligned} \quad (B.41)$$

$$\underline{M17 = N17}$$

$$\begin{aligned} -d_6 \sin (\theta_2 + \theta_3 + \theta_4) \sin \theta_5 - d_5 \cos (\theta_2 + \theta_3 + \theta_4) + a_3 \sin (\theta_2 + \theta_3) \\ + a_2 \sin \theta_2 + d_1 = z \end{aligned} \quad (B.42)$$

Apéndice C

Productos de transformación para Hombro a muñeca 1

Esta parte muestra los productos de transformación para la estructura cinemática del robot UR3 a partir del marco 2 (hombro) al marco 4 (muñeca 1).

Definimos el producto dual de cuaterniones M_i como:

$$M_i = Q_i \quad M_{i+1} \text{ donde } 2 \leq i \leq 4$$

$$M_4 = Q_4$$

$$M_3 = Q_3 \quad M_4 = Q_3 \quad Q_4$$

$$M_2 = Q_2 \quad M_3 = Q_2 \quad Q_3 \quad Q_4$$

Definimos el producto de cuaternión dual N_i de la siguiente manera:

$$N_{i+1} = Q^{-1}_i \quad N_i \text{ donde } 2 \leq i \leq 4$$

$$N_2 = Q_2^{-1} \quad ([w_4, < a_4, b_4, c_4 >], < x_4, y_4, z_4 >)$$

$$N_3 = Q^{-1}_2 \quad N_2 = Q^{-1}_2 \quad Q_2 \quad Q_4$$

$$N_4 = Q^{-1}_3 \quad N_3 = Q^{-1}_3 \quad Q^{-1}_2 \quad Q_2 \quad Q_4$$

Apéndice C. Productos de transformación de hombro a muñeca 1

Productos de transformación de hombro a muñeca 1 ($M4 = N4$)

$$M4 = ([M41 + M42i + M43j + M44k], < M45i + M46j + M47k >)$$

$$N4 = ([N41 + N42i + N43j + N44k], < N45i + N46j + N47k >)$$

$$\underline{M41 = N41}$$

$$\text{porque } \frac{\theta 4}{2} = w4 \text{ porque } \frac{\theta 2 + \theta 3}{2} - b4 \text{ pecado } \frac{\theta 2 + \theta 3}{2} \quad (C.1)$$

$$\underline{M42 = N42}$$

$$0 = a4 \text{ porque } \frac{\theta 2 + \theta 3}{2} + c4 \text{ pecado } \frac{\theta 2 + \theta 3}{2} \quad (C.2)$$

$$\underline{M43 = N43}$$

$$- \text{pecado } \frac{\theta 4}{2} = w4 \text{ pecado } \frac{\theta 2 + \theta 3}{2} + b4 \text{ porque } \frac{\theta 2 + \theta 3}{2} \quad (C.3)$$

$$\underline{M44 = N44}$$

$$0 = -a4 \text{ sen } \frac{\theta 2 + \theta 3}{2} + c4 \text{ porque } \frac{\theta 2 + \theta 3}{2} \quad (C.4)$$

$$\underline{M45 = N45}$$

$$0 = -a2 \cos \theta 3 - a3 + x4 \cos (\theta 2 + \theta 3) + z4 \text{ sen } (\theta 2 + \theta 3) \quad (C.5)$$

$$\underline{M46 = N46}$$

$$0 = y4 + d4 \quad (C.6)$$

$$\underline{M47 = N47}$$

$$0 = a2 \text{ sen } \theta 3 - x4 \text{ sen } (\theta 2 + \theta 3) + z4 \cos (\theta 2 + \theta 3) \quad (C.7)$$

Productos de transformación de hombro a muñeca 1 ($M3 = N3$)

$$M3 = ([M31 + M32i + M33j + M34k], < M35i + M36j + M37k >)$$

$$N3 = ([N31 + N32i + N33j + N34k], < N35i + N36j + N37k >)$$

$$\underline{M31 = N31}$$

$$\text{porque } \frac{\theta_3 + \theta_4}{2} = w_4 \text{ porque } \frac{\theta_2}{2} - b_4 \text{ pecado } \frac{\theta_2}{2} \quad (C.8)$$

$$\underline{M32 = N32}$$

$$0 = a_4 \text{ porque } \frac{\theta_2}{2} + c_4 \text{ pecado } \frac{\theta_2}{2} \quad (C.9)$$

$$\underline{M33 = N33}$$

$$- \text{pecado } \frac{\theta_3 + \theta_4}{2} = w_4 \text{ pecado } \frac{\theta_2}{2} + b_4 \text{ porque } \frac{\theta_2}{2} \quad (C.10)$$

$$\underline{M34 = N34}$$

$$0 = -a_4 \text{ sen } \frac{\theta_2}{2} + c_4 \text{ porque } \frac{\theta_2}{2} \quad (C.11)$$

$$\underline{M35 = N35}$$

$$a_3 \cos \theta_3 = -a_2 + x_4 \cos \theta_2 + z_4 \text{ sen } \theta_2 \quad (C.12)$$

$$\underline{M36 = N36}$$

$$-d_4 = y_4 \quad (C.13)$$

$$\underline{M37 = N37}$$

$$a_3 \text{ sen } \theta_3 = -x_4 \text{ sen } \theta_2 + z_4 \cos \theta_2 \quad (C.14)$$

Apéndice C. Productos de transformación de hombro a muñeca 1

Productos de transformación de hombro a muñeca 1 ($M2 = N2$)

$$M2 = ([M21 + M22i + M23j + M24k], < M25i + M26j + M27k >)$$

$$N2 = ([N21 + N22i + N23j + N24k], < N25i + N26j + N27k >)$$

$$\underline{M21 = N21}$$

$$\text{porque } \frac{\theta_2 + \theta_3 + \theta_4}{2} = w_4 \quad (C.15)$$

$$\underline{M22 = N22}$$

$$0 = a_4 \quad (C.16)$$

$$\underline{M23 = N23}$$

$$\text{- pecado } \frac{\theta_2 + \theta_3 + \theta_4}{2} = b_4 \quad (C.17)$$

$$\underline{M24 = N24}$$

$$0 = c_4 \quad (C.18)$$

$$\underline{M25 = N25}$$

$$a_3 \cos(\theta_2 + \theta_3) + a_2 \cos \theta_2 = x_4 \quad (C.19)$$

$$\underline{M26 = N26}$$

$$-d_4 = y_4 \quad (C.20)$$

$$\underline{M27 = N27}$$

$$a_3 \sin(\theta_2 + \theta_3) + a_2 \sin \theta_2 = z_4 \quad (C.21)$$

Apéndice D

Lista de tablas y figuras

Lista de tablas

3.1 Tabla de multiplicación de cuaterniones.	27
4.1 Parámetros de UR3 Denavit-Hartenberg.	43
4.2 Parámetros Denavit-Hartenberg para UR3 de [20]	57

Lista de Figuras

2.1 Representación vectorial de $z = a + ib$ con $\text{Re}z = a$ e $\text{Im}z = b$.	17
2.2 Representación en coordenadas polares del número complejo $z = a + ib$ con longitud r y ángulo θ .	17
2.3 Conjugación compleja.	18
2.4 Suma de números complejos $3 + 2i$ y $1 + 3i$.	20
2.5 Multiplicación de números complejos z_1 y z_2 .	21
2.6 Multiplicación de números complejos -1 y $-i$.	21
2.7 Multiplicación de wz donde $z = i$.	22
3.1 Un retrato de Sir William Rowan Hamilton ¹ .	24
3.2 Placa conmemorativa del descubrimiento de los cuaterniones. ²	24
3.3 Diagrama de multiplicación de cuaterniones.	27
3.4 Rotación del vector $(2,1,1)$ de [22].	35
4.1 Robot Universal UR3.	41
4.2 Estructura Cinemática de UR3 en posición cero ($\theta_1, 2, 3, 4, 5, 6 = 0$) [3].	42
4.3 Marcos de coordenadas de UR3 ($\theta_1, 2, 3, 4, 5, 6 = 0$) de [11].	47
4.4 Estructura del robot UR3 del marco 1 al marco 5 (desde [3]).	48
4.5 Traslación del Marco 6 (Muñeca 3) al Marco 5 (Muñeca 2)[3].	49
4.6 Manipulador 3R-Planar formado por UR3 Articulaciones 2 a 4 de [3].	53
4.7 Diagrama de parámetros UR3 DH de [20].	57

Bibliografía

- [1] Robótica: Cinemática y Fundamentos Matemáticos.
<https://www.edx.org/course/robotics-kinematics-mathematical-pennx-robo1x>.
- [2] M. Ahr. El legado de los robots universales de Rossum.
<https://www.denofgeek.com/us/culture/279175/the-legacy-of-rossums-universal-robots>, 2019.
- [3] R. Andersen. Cinemática de un UR5. Informe técnico, Universidad de Aalborg ciudad, 2018.
- [4] Y. Aydin y S. Kucuk. Cinemática inversa basada en cuaterniones para manipuladores de robots industriales con muñeca de Euler. 2006 IEEE Internacional Conferencia sobre Mecatrónica, ICM, 2006.
- [5] J. Barbic. Cuaterniones y Rotaciones. Informe técnico, Universidad de Sur de California, 2011.
- [6] M. Ben-Ari. Un tutorial sobre ángulos de Euler y cuaterniones. 2018.
- [7] Z. Chen y J. Hung. Aplicación de Quaternion en Control de Robots. Décimo Congreso Mundial Trienal de la IFAC, 1987.
- [8] J. Diebel. Representación de la actitud: ángulos de Euler, cuaterniones unitarios y Vectores de rotación. Informe técnico, Universidad de Stanford, 2006.
- [9] PR Evans. Rotaciones y Matrices de Rotación. Informe técnico, MRC Laboratorio de Biología Molecular, 2001.
- [10] WR Hamilton. Elementos de cuaterniones. Prensa de la Universidad de Cambridge, 1866.
- [11] K. Hawkins. Cinemática inversa analítica para Universal Robots UR 5/UR-10 Brazos. Informe técnico, 2013.

Bibliografía

- [12] Colina AO. Cinemática: por qué los robots se mueven como lo hacen. [https://blog.robotiq.com/kinematics-why-robots-move-like-they do](https://blog.robotiq.com/kinematics-why-robots-move-like-they-do), 2015.
- [13] J.Huerta. Introducción a los cuaterniones. Informe técnico, Fullerton College, 2010.
- [14] B. Kenwright. Una guía para principiantes de cuaterniones duales: qué son, cómo funcionan y cómo usarlos para 3D. Informe técnico, Universidad de Newcastle, Reino Unido.
- [15] S. Kucuk y Z. Bingul. Cinemática de robots: cinemática directa e inversa. Informe técnico, Universidad de Kocaeli, 2006.
- [16] S. LaValle. Algoritmos de Planificación - La Matriz de Transformación Homogénea. <http://planning.cs.uiuc.edu/node111.html>, 2006.
- [17] A. Lieros. Rotaciones y Cuaterniones. Informe técnico, Universidad de Stanford Versity, 1995.
- [18] JE Marsden y MJ Hoffman. Análisis complejo básico. WHFreeman & Co Ltd, Nueva York, Estados Unidos, 3ª edición, 1999.
- [19] Mataric. El manual de robótica. Instituto de Tecnología de Massachusetts, Londres, Inglaterra, 2007.
- [20] U. Robots. Parámetros para Cálculos de Cinemática y Dinámica. <https://www.universal-robots.com/how-to-and-faq/faq/ur-faq/parameters-for-calculations-of-kinematics-and-dynamics-45257/>.
- [21] B. Siciliano y O. Khatib. Manual Springer de Robótica. Saltador, Berlín, 2ª edición, 2016.
- [22] M. Sunardi. Cinemática 3D. Informe técnico, 2006.
- [23] Tekkotsu. Cinemática. <http://www.tekkotsu.org/Kinematics.html>, 2010.
- [24] J. Voight. Quaternion Algebras, volumen 0.9.14. 2018.
- [25] A. Watt y M. Watt. Técnicas Avanzadas de Animación y Renderizado. ACM Press, Nueva York, Estados Unidos, 1992.