

ANÁLISIS DE TRAYECTORIAS Y SU SIMULACIÓN EN MATLAB UTILIZANDO DIVERSAS TÉCNICAS DE INTERPOLACIÓN LINEAL

Ramos Pedro¹ Caraguay Washington²

Resumen. Este artículo presenta un análisis de la generación y simulación de trayectorias producto del movimiento rotacional del sistema de referencia de la orientación entre un punto inicial y un punto final. Estos puntos pueden ser establecidos por matrices de rotación, ángulos de Euler, cuaternios o pares de rotación, que haciendo uso de técnicas de interpolación lineal generan una trayectoria, la misma que luego es mapeada sobre una esfera unitaria. Para la simulación se desarrolla una interfaz gráfica en MatLab, donde se visualiza el movimiento del sistema de referencia móvil y la trayectoria generada entre los puntos.

Palabras claves: Sistema de Referencia, orientación, interpolación lineal, trayectoria, esfera unitaria.

Abstract. This paper presents an analysis of the generation and simulation of trajectories as result of rotational movement in the orientation reference between a start point and an end point. These points can be established by rotation matrices, Euler's angles, quaternions or rotation's pairs that using linear interpolation techniques generate a trajectory and then it is mapped onto a unit sphere. To simulate the trajectories has been developed a graphic interface in MatLab, where the movement of the mobile reference system and the trajectory generated between the points is displayed.

Keywords: Reference system, orientation, linear interpolation, trajectory, unit sphere.

Recibido: Julio 2014

Aceptado: Agosto 2014

1. INTRODUCCIÓN

Un punto queda definido en el espacio a través de los datos de su posición. Sin embargo, la localización de un cuerpo rígido precisa de especificación tanto de posición como de orientación. Si estos principios son aplicados a las tareas de manipulación de un robot industrial, es necesario conocer la posición y orientación de los elementos a manipular con referencia a su base. Una orientación en el espacio tridimensional está definida por tres grados de libertad o tres componentes linealmente independientes. Entre las herramientas más habituales para representar puntos de orientación contamos con matrices de rotación, ángulos de Euler, pares de rotación y cuaternios. Estas herramientas son equivalentes pero su elección depende de la aplicación en que se requiera trabajar.

En este artículo, haciendo uso de diversas técnicas de interpolación lineal, se analiza la evolución de la orientación entre dos puntos, para lo cual se mapea la trayectoria generada sobre la superficie de una esfera unitaria.

Esta interpolación se origina con un valor inicial y termina en un valor final. El análisis de los resultados obtenidos está en función de la trayectoria generada que realice la técnica de interpolación y la fórmula matemática utilizada para representar los puntos de la orientación.

Para la visualización de estas trayectorias se desarrolla una interfaz gráfica en MatLab, donde se pueden configurar diferentes modos de representar los puntos de orientación y adicionalmente visualizar el movimiento rotacional del sistema de referencia y la velocidad con que es llevada la trayectoria.

2. BASES CONCEPTUALES

Rotación.

La rotación en 3D no es tan simple como la traslación y puede ser definida de muchas formas. La definición dada por el teorema de Euler (Euler, 1752) y escrita en notación moderna, dice: Sean $O, O' \in \mathbb{R}^3$ dos orientaciones, entonces existe un eje $l \in \mathbb{R}^3$ y un ángulo de rotación $\theta \in [-\pi, \pi]$ tales que O produce O' cuando se hace rotar θ alrededor de l . Note que esta proposición expresa existencia y no unicidad. Es importante distinguir entre orientaciones y rotaciones. Una orientación de un objeto en R^3 está dada por un vector normal. Una rotación está definida por un eje y un ángulo de rotación.

Ángulos de Euler.

El espacio de las orientaciones puede ser parametrizado por medio de los ángulos de Euler. Cuando son utilizados, una orientación general se escribe como una serie de rotaciones alrededor de tres ejes mutuamente ortogonales en un espacio determinado. Usualmente se utilizan los ejes x, y, z en un sistema cartesiano y las rotaciones pasan a llamarse giro- x , giro- y , giro- z .

¹ Ramos Pedro, M.Sc., Profesor, Universidad Espíritu Santo-Ecuador. (Email: pedroramos@uees.edu.ec)

² Caraguay Washington, M.Sc., Profesor, Universidad Espíritu Santo-Ecuador. (Email: wcaraguay@uees.edu.ec)

Euler originalmente desarrolló los ángulos de Euler como una fórmula para resolver ecuaciones diferenciales. Luego, los ángulos de Euler se convirtieron en el método más ampliamente utilizado para la parametrización de espacios de orientaciones. Si se elige considerar una rotación como la acción desarrollada para obtener el espacio de rotaciones, los ángulos de Euler pueden ser utilizados para parametrizar el espacio de rotaciones. Para describir una rotación general se requieren 3 ángulos de Euler $(\theta_1, \theta_2, \theta_3)$, donde θ_1, θ_2 y θ_3 son los ángulos de rotación alrededor de los ejes x, y, z respectivamente. La conversión de una rotación general a ángulos de Euler es ambigua desde que la misma rotación puede ser obtenida con diferentes conjuntos de ángulos de Euler (Foley et al., 1990). Además la rotación resultante depende del orden en que las tres rotaciones son desarrolladas.

Matrices de rotación.

Las matrices de rotación son la elección típica para implementar ángulos de Euler. Para cada tipo de rotación, hay una correspondiente matriz de rotación. Una matriz de rotación es una matriz 3x3, pero también matrices homogéneas 4x4 pueden ser utilizadas en su lugar (Foley et al., 1990). Una rotación general es obtenida multiplicando las tres matrices de rotación correspondientes a los tres ángulos de Euler. La matriz resultante incorpora la rotación general y puede ser aplicada a los puntos que van a ser rotados.

La matriz multiplicación generalmente no es conmutativa, esto encaja bien con el hecho de que las rotaciones en el espacio no conmutan.

Cuaternios.

Los cuaternios fueron inventados por Sir William Rowan Hamilton. Su objetivo fue generalizar los números complejos a tres dimensiones, por ejemplo, números de la forma $a + ib + jc$, donde $a, b, c \in R$ y se cumple que $i^2 = j^2 = -1$. Hamilton nunca tuvo éxito en hacer esta generalización. Más tarde, en 1996, Kenneth O. May probó que el conjunto de números de tres dimensiones no es cerrado bajo la multiplicación.

Para expresar la matemática básica de los cuaternios, se usa la siguiente notación:

\equiv Significa igualdad por definición;
 $[a, b] \equiv \{x/a \leq x \leq b\}, a, c, x \in \mathbb{R}$ significa intervalo cerrado en la recta real;
 $[a, b) \equiv \{x/a < x < b\}, a, c, x \in \mathbb{R}$ significa un intervalo semi-abierto.

El conjunto de cuaternios se denota por H . Los cuaternios consisten de una parte escalar $s \in \mathbb{R}$ y una parte vectorial $(x, y, z) \in R^3$.

Sea $i^2 = j^2 = k^2 = ijk = -1, ij = k$ y $ji = -k$, luego $q \in H$, puede ser escrito como:

$$q \equiv [s, \mathbf{v}], s \in R, \mathbf{v} \in \mathbb{R}^3$$

$$\equiv [s, (x, y, z)], s, x, y, z \in \mathbb{R}$$

$$\equiv s + ix + jy + kz, s, x, y, z \in R$$

Se identifica el conjunto de cuaternios $\{[s, \mathbf{0}]/s \in \mathbb{R}\}$ con \mathbb{R} y el conjunto $\{[0, \mathbf{v}]/\mathbf{v} \in \mathbb{R}^3\}$ con \mathbb{R}^3 .

La multiplicación de cuaternios no es generalmente conmutativa, pero si es asociativa y distributiva con respecto a la suma. La multiplicación de cuaternios por un escalar se introduce más fácilmente identificando a $r \in \mathbb{R}$ con el cuaternio $[r, \mathbf{0}]$.

Sean $q \in H$ y $r \in \mathbb{R}$, la multiplicación por un escalar se define como $rq \equiv [r, \mathbf{0}]$.

Sea $q \in H$, entonces q^* se denomina el conjugado de q y se define como $q^* = [s, \mathbf{v}]^* = [s, -\mathbf{v}]$.

Sea $p \in H$ y sea el trazado $\|\cdot\|: H \rightarrow \mathbb{R}$, definido por $\|q\| = \sqrt{qq^*}$, este trazado es denominado la norma y $\|q\|$ es la norma de q . Sean $q, q' \in H$, $q = [s, \mathbf{v}] = [s, (x, y, z)], q' = [s', \mathbf{v}'] = [s', (x', y', z')]$. El producto interno se define como $\cdot: H \times H \rightarrow \mathbb{R}$ donde $q \cdot q' = ss' + \mathbf{v} \cdot \mathbf{v}' = ss' + xx' + yy' + zz'$.

El conjunto de cuaternios $H \setminus \{[0, (0,0,0)]\}$ se denota con \dot{H} y es un grupo No-Abeliano bajo la multiplicación de cuaternios.

Sea G un conjunto con un operador: $G \times G \rightarrow G$ definido por $(a, b) \rightarrow a \cdot b \equiv ab$. G es un grupo, si:

- i) $a(bc) = (ab)c; \forall a, b, c \in G$. El operador es asociativo.
- ii) $\exists I \in G; Ia = aI; \forall a \in G$. I es el elemento neutro.
- iii) $\forall a \in G \exists a^{-1} \in G; aa^{-1} = a^{-1}a = I$. a^{-1} es el elemento inverso de a .

Si $ab = ba; \forall a, b \in G$, entonces G se denomina un grupo Abelian o conmutativo.

Rotación con cuaternios.

Hamilton trató de describir las rotaciones en el espacio solo como números complejos que describen rotaciones en el plano y en efecto, los cuaternios desarrollan rotación.

Sea $p \in H, p = [s, (x, y, z)] = (s, \mathbf{v})$ y sea $q \in \dot{H}$. Si $r \in \mathbb{R} \setminus \{0\}$, entonces $(rq)p(rq)^{-1} = qpq^{-1}$.

Sea $q \in H_1, p = [s, \mathbf{v}] \in H$, entonces $qpq^{-1} = p'$, donde $p' = [s, \mathbf{v}']$ con $|\mathbf{v}| = |\mathbf{v}'|$.

Sea $q \in H_1, p = [a, b\mathbf{v}] \in H$, donde $a, b \in R$ y $\mathbf{v} \in \mathbb{R}^3, y = [a, \mathbf{v}']$, entonces $q[a, \mathbf{v}]q^* = [a, b\mathbf{v}']$.

Sean $q, p \in H_1, p = [\cos\theta, \text{sen}\theta\mathbf{v}]$, $t \in \mathbb{R}$, entonces $qpq^t q^* = (qpq^*)^t$.

En base a lo anterior, se puede mostrar el teorema principal inspirado por (Watt & Watt, 1992): Sea $q \in H_1, q = [\cos\theta, \text{sen}\theta\mathbf{n}]$. Sea $\mathbf{r} = (x, y, z) \in \mathbb{R}^3$ y $p = [0, \mathbf{r}] \in H$, entonces $p' = qpq^{-1}$ es p rotado 2θ alrededor del eje \mathbf{n} .

Cualquier rotación tridimensional general θ alrededor de $\mathbf{n}, |\mathbf{n}|=1$, puede ser obtenida por un cuaternio unitario.

Sean $q_1, q_2 \in H_1$, la rotación de q_1 seguida por la rotación de q_2 es equivalente a la rotación de q_2q_1 .

Interpolación Lineal de Euler.

El método más obvio para la interpolación de la rotación es la simple interpolación lineal entre dos componentes de ángulos de Euler. Se denota a ésta interpolación curva *LinEuler*, la interpolación entre $v_0 = (x_0, y_0, z_0) \in \mathbb{R}^3$ y $v_1 = (x_1, y_1, z_1) \in \mathbb{R}^3$ puede ser expresada algorítmicamente usando $h \in [0,1]$ como el parámetro de interpolación:

$$LinEuler(v_0, v_1, h) = v_0(1 - h) + v_1h$$

Interpolación Lineal con matrices de rotación.

Una alternativa simple es la interpolación lineal entre las matrices de rotación, es decir, la interpolación de cada elemento de cada matriz independientemente de las otras.

Esto puede ser expresado logarítmicamente de manera sencilla. Con el parámetro $h \in [0,1]$, la curva de interpolación entre las matrices de rotación $M_0 \in \mathbb{R}^4 \times \mathbb{R}^4$ y $M_1 \in \mathbb{R}^4 \times \mathbb{R}^4$, la curva se define como:

$$LinMat(M_0, M_1, h) = M_0(1 - h) + M_1h$$

Las matrices interpoladas son matrices de transformación generalmente homogéneas que contienen traslación, escalación, proyección y otros elementos de transformación. Por lo tanto, la interpolación puede volverse arbitrariamente errónea. Por ejemplo, es posible colapsar todo el objeto en un simple punto.

Interpolación Lineal con Cuaternios.

Otra opción es la interpolación lineal entre cuaternios de rotación. Para $q_0, q_1 \in H$ y $h \in [0,1]$, esta curva de interpolación puede ser expresada como:

$$Lerp(q_0, q_1, h) = q_0(1 - h) + q_1h$$

La curva de interpolación para interpolación lineal entre cuaternios entrega una línea recta en el espacio de cuaternios. Por eso, la curva cae sobre la superficie de la esfera unitaria y como todos los cuaternios sobre una línea a través del origen, entregan la misma rotación; la curva puede ser proyectada sobre la esfera unitaria sin que cambien las rotaciones correspondientes. Por esta razón, la curva de interpolación es normalizada.

Interpolación Lineal Esférica con Cuaternios.

A pesar de que todos los cuaternios sobre una línea a través del origen producen la misma rotación, es de interés en este caso solo utilizar cuaternios para rotación, ya que poseen un rango deseable de propiedades. La interpolación lineal simple con cuaternios produce una secante entre dos cuaternios, por eso la función interpolación tiene una velocidad más alta arco sobre la esfera unitaria desde una estructura a otra. Esta interpolación se denomina interpolación de gran arco o interpolación lineal esférica *Slerp*.

$$\begin{aligned} Slerp(p, q, h) &= (pp^*q)^h \\ Slerp(p, q, h) &= (pq^*)^{1-h}q \\ Slerp(p, q, h) &= (qp^*)^hp \end{aligned}$$

$$Slerp(p, q, h) = q(q^*p)^{1-h} \text{ (Shoemake, 1987).}$$

Spline esférico en cuaternios.

A pesar de que esta opción no se incluye en la aplicación desarrollada en MatLab, se describe a continuación su fundamentación matemática: esta curva de interpolación es llamada *Squad* (spherical and quadrangle) y fue presentada en el año 1987 por Shoemake.

$$\begin{aligned} Squad(q_i, q_{i+1}, h_i, h_{i+1}, h) &= \\ Slerp \left(\begin{array}{c} Slerp(q_i, q_{i+1}, h), Slerp(s_i, s_{i+1}, h) \\ 2h(1 - h) \end{array} \right) \\ s_i &= q_i \exp \left(-\frac{\log(q_i^{-1}q_{i+1}) + \log(q_i^{-1}q_{i-1})}{4} \right) \end{aligned}$$

La definición de *Squad* es compleja y por eso ni la continuidad ni la diferenciabilidad de la curva de interpolación resultante es obvia. *Squad* ha servido como referencia general para la prueba de diferenciabilidad de *Squad*. Luego se presentó una nueva prueba (Kim et al, 1996).

La expresión algorítmica para *Squad* produce una curva de interpolación para una serie de cuaternios q_0, \dots, q_N . La expresión no está definida en el primer y último intervalos dado que q_{-1} aparece en la expresión para s_0 y q_{N+1} aparece en la expresión para s_N . La solución más simple es definir $s_0 \equiv q_0$ y $s_N \equiv q_N$, alternativamente q_{-1} y q_{N+1} también pueden ser definidos.

3. DESCRIPCIÓN DE LA APLICACIÓN Y ANÁLISIS DE RESULTADOS

La aplicación está desarrollada en MatLab, en entorno gráfico. Se desarrollan funciones para los tipos de interpolaciones descritos, así como también para las diferentes conversiones de la orientación. La figura 1 muestra una imagen de la interfaz gráfica. Se utilizan los valores predeterminados que desarrolla el software, los mismos que son los puntos de orientaciones tanto inicial como final, los cuales se detallan en la tabla 1. Cabe mencionar que son equivalentes debido a la aplicación de sus relaciones de conversión.

Interpolación Lineal en Ángulos de Euler WUW y RPY.

Al trabajar con Matrices de Rotación corremos el riesgo de que estas Matrices no sean ortogonales por lo que no se va a efectuar evolución alguna. De ser así no estaría permitido realizar una interpolación lineal con matrices de rotación ya que involucraría a orientaciones erróneas del extremo del robot. Sin embargo se puede hacer uso de la conversión de matrices de rotación a ángulos de Euler y superar en parte el problema. Los valores predefinidos en la tabla 1, involucran una orientación final con una matriz de rotación que no es ortogonal. La figura 3 muestra la solicitud errónea al ejecutar esta acción.

FIGURA 1

Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal
Imagen de la interfaz gráfica desarrollada

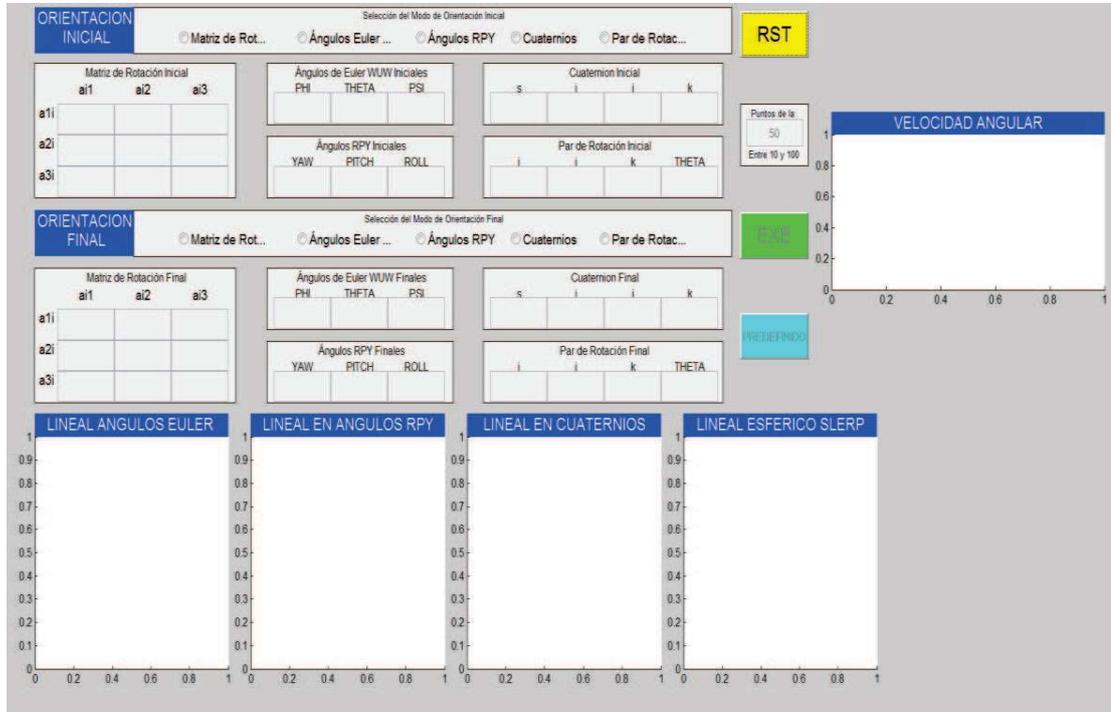


TABLA I

Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal
Datos de orientaciones iniciales y finales

	Matrices de Rotación	Ángulos de Euler WUW (ϕ, θ, ψ)	Ángulos de Euler RPY (ψ, θ, ϕ)	Cuaternios (s, q_1, q_2, q_3)	Pares de Rotación (i, j, k, θ)
Orientación Inicial	$R_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$E_{wuw_i} = (0, 0, 0)$	$E_{rpy_i} = (0, 0, 0)$	$q_i = (1, 0, 0, 0)$	$park_i = (0.0707, 0.7071, 0.7071, 0)$
Orientación Final	$R_f = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\sqrt{2}}{2} & 1 & -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \end{bmatrix}$	$E_{wuw_f} = (0, 2.3562, -\frac{\pi}{2})$	$E_{rpy_f} = (\frac{\pi}{4}, \frac{\pi}{2})$	$q_f = (0.270, 0.653, 0.653, -0.270)$	$park_f = (0.6786, 0.6786, -0.2811, 2.5936)$

La utilización de los ángulos de Euler, en cualquiera de sus formas, no presenta este inconveniente. Así, podemos utilizar entonces la fórmula de la interpolación lineal aplicada a los ángulos de Euler, la misma que consiste en:

$$\theta(t) = (\theta_f - \theta_i)t + \theta_i; \quad \phi(t) = (\phi_f - \phi_i)t + \phi_i$$

Sin embargo uno de los principales inconvenientes desde el punto de vista del usuario, es que se trata de una trayectoria no intuitiva, con extrañas evoluciones de la orientación. Con la ecuación de interpolación lineal para ángulos de Euler, en cierto modo, se asegura la continuidad de la posición del punto de orientación, sin embargo origina saltos bruscos en la velocidad de la articulación, y en consecuencia la presencia de aceleraciones de valor infinito. La figura 3 muestra la interpolación realizada por este interpolador tanto para ángulos de

FIGURA 2
Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal
Error esperado al introducir matrices de rotación no ortonormales

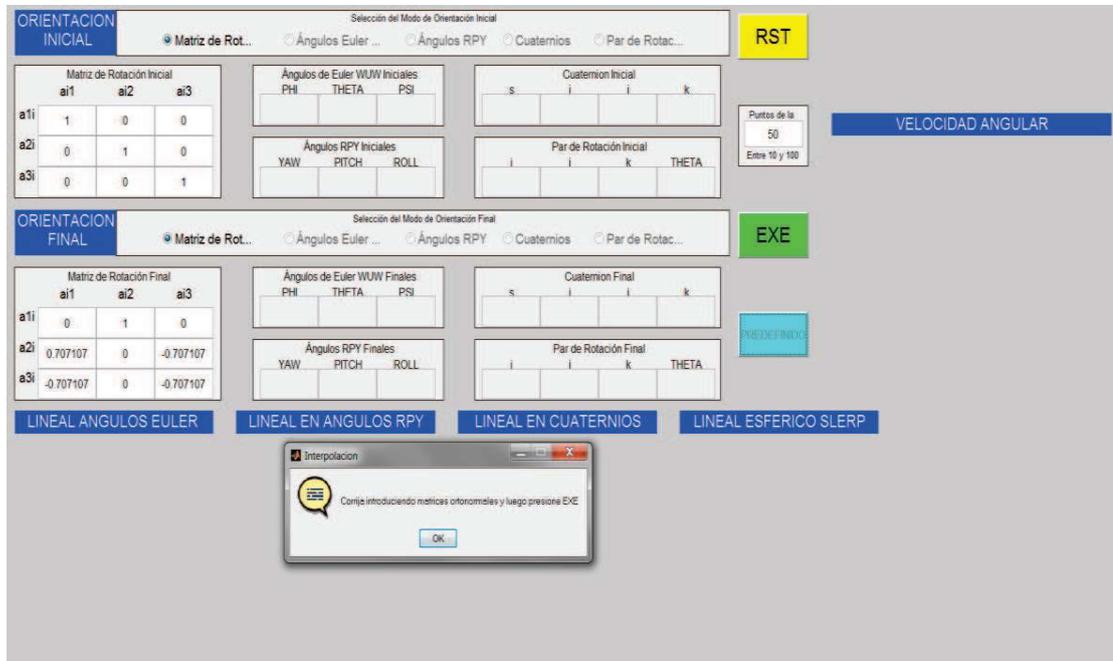
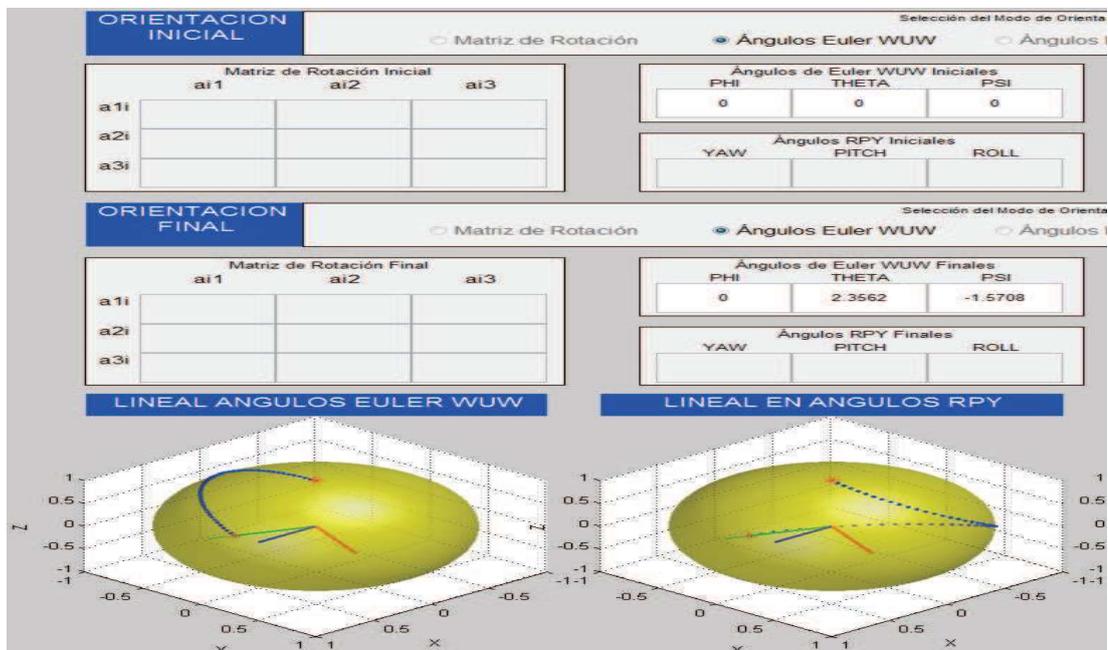


FIGURA 3
Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal
Evolución de la orientación al utilizar interpolador lineal en ángulos de Euler



Euler WUW como para ángulos RPY considerando 50 puntos en total.

Otros de los puntos a destacar son las velocidades angulares que se generan al realizar la trayectoria interpoladora. Para su implementación se utiliza la siguiente fórmula como una aproximación a su velocidad:

$$V(q_i) = \frac{\|q_i - q_{i-1}\| + \|q_i - q_{i+1}\|}{2}$$

Que no es más que la velocidad lineal entre un cuaternio inicial y uno final. Se puede observar que la velocidad para la interpolación lineal en ángulos de Euler WUW es constante (figura 4), a diferencia de la velocidad obtenida por la interpolación lineal en ángulos RPY (figura 5). Los ángulos de Euler por su naturaleza, presentan ciertas singularidades, como por ejemplo, cuando se pierde un grado de libertad (efecto gimbal lock), consecuentemente se originan ambigüedades en la correspondencia de rotaciones. La figura 7 muestra esta condición al introducir Matrices de Rotación.

Interpolación Lineal en Cuaternios LERP.

Es similar a la anterior pero con la diferencia que lo que se interpola no son ángulos de Euler sino cuaternios (figura 6) y está definida por la siguiente ecuación:

$$\text{Lerp}(q_0, q_1, t) = q_0(1 - t) + q_1 t$$

Cuando se utilizan cuaternios, se simplifica mucho el cálculo de la composición de rotaciones, por ejemplo la rotación de q_1 , seguida por la rotación con q_2 , es lograda simplemente por la operación de multiplicación entre cuaternios, es decir, $q_2 * q_1$, lo que lo convierte en un método computacionalmente muy práctico. Cuando se trabaja con cuaternios no se presentan problemas de gimbal lock.

Dado que todos los cuaternios en una recta a través del origen dan la misma rotación, la curva puede ser proyectada sobre la esfera unitaria sin cambiar el giro correspondiente.

Aunque la curva proyectada de la Interpolación Slerp es muy conveniente, como se muestra en la figura 8, no ocurre lo mismo con la de su velocidad, ya que la curva transita por debajo de la esfera unitaria, lo cual se puede observar en la figura 9. Lo que se requiere es una gráfica de velocidad que sea constante en todos los puntos de la trayectoria interpolada.

Interpolación Lineal Esférico en Cuaternios SLERP.

A diferencia de LERP, el movimiento generado con SLERP es constante, esto se debe a que en lugar de realizar interpolación lineal simple, la curva sigue un arco sobre la esfera unitaria desde una orientación inicial a una final. Este arco también es conocido como la geodésica de la línea recta de la interpolación Slerp. La fórmula usada para este

método es la siguiente:

$$\text{Slerp}(q_0, q_1, t) = q_0 \left(\frac{\sin(1-t)\theta}{\sin\theta} \right) + q_1 \left(\frac{\sin(t\theta)}{\sin\theta} \right)$$

Donde $t \in [0,1]$ y θ es el ángulo entre q_0 y q_1 y está dado por:

$$\theta = \cos^{-1}(q_0 * q_1)$$

La figura 10 muestra la gráfica de la trayectoria interpolada por Slerp, considerando como orientaciones iniciales y finales a los cuaternios y 100 puntos a interpolar. La característica de una interpolación Slerp es que su velocidad es constante, como se muestra en la figura 11.

FIGURA 4

Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal

Gráfico de velocidad para ángulos WUW



FIGURA 5

Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal

Gráfico de velocidad para ángulos RPY

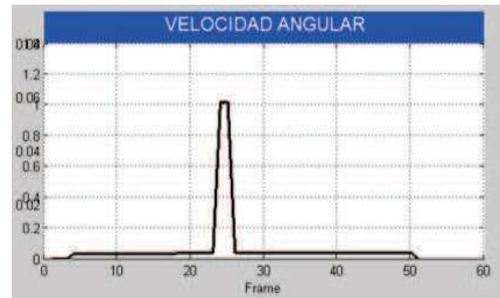


FIGURA 6

Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal
Curva de interpolación Lerp

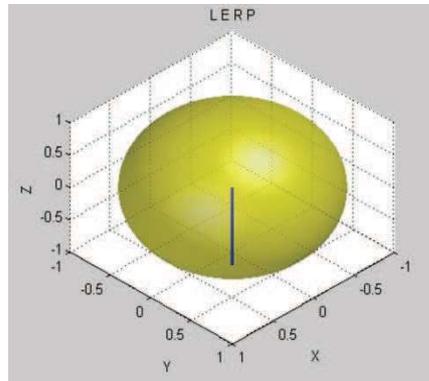


FIGURA 7

Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal
Ambigüedades producidas por la interpolación con ángulos de Euler

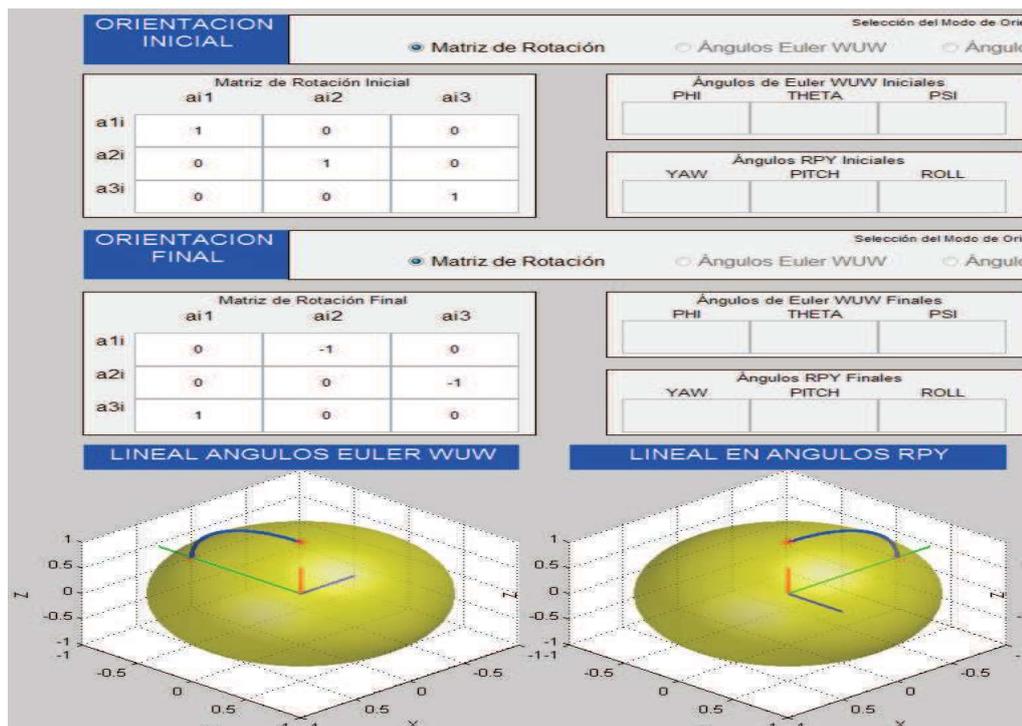


FIGURA 8

Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal

Curva de interpolación Lerp

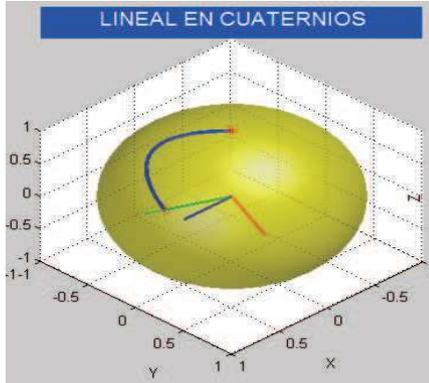


FIGURA 9

Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal

Velocidad de interpolación Lerp

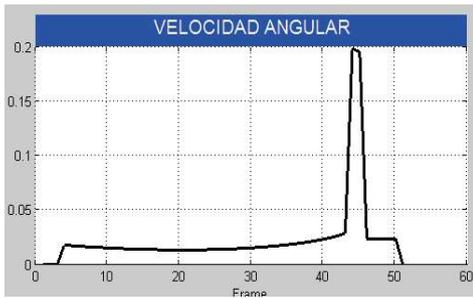


FIGURA 10

Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal

Trayectoria interpolada Slerp

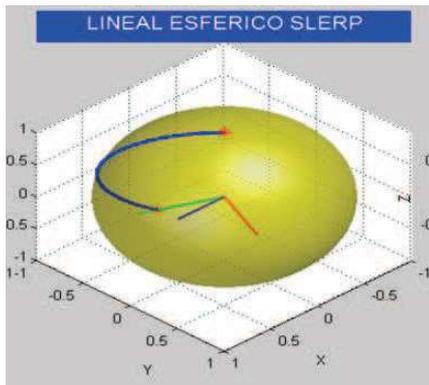


FIGURA 11

Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal

Velocidad de interpolación Slerp



4. CONCLUSIONES Y CONSIDERACIONES FINALES

Se ha desarrollado una interfaz gráfica que muestra la evolución del punto desde una orientación inicial hasta una orientación final, proyectándose la trayectoria interpolada sobre una esfera unitaria. Una captura de pantalla completa de la aplicación se muestra en la figura 12.

Todos los puntos de orientación que se consideren pueden ser definidos en matrices de rotación, ángulos de Euler y pares rotacionales.

Más allá del análisis de los diferentes tipos de interpoladores para la trayectoria interpolada en n puntos, que se tratan en este artículo, tales como la interpolación lineal en ángulos de Euler, Lerp y Slerp, es importante anotar que las curvas de interpolación en el plano han inspirado varias curvas de interpolación para rotaciones, siendo las más importantes y de mayor estudio Slerp y la mencionada en el presente artículo, pero no incluida en la aplicación en MatLab, llamada Squad.

Los ángulos de Euler y las matrices de rotación son las modalidades más comúnmente utilizados tanto en la literatura pertinente como en las aplicaciones, sin embargo este trabajo muestra que existe otra opción llamada interpolación lineal esférica en cuaternios y conocida como Slerp, que resulta ser más apropiada.

Del análisis realizado, resaltan las desventajas de la interpolación lineal de Euler (LinEuler) y de la interpolación lineal con matrices de rotación (LinMat), ya que los ángulos de Euler no son la mejor definición existente para rotación, razón por la cual no debe esperarse que la simple interpolación lineal entre pares de ángulos de Euler y matrices de rotación resulten en buenas curvas de interpolación.

La curva de interpolación para Slerp es bastante mejorada, siendo su único problema la variación en la velocidad. Pero hay que enfatizar que

dependiendo de la aplicación a utilizar, quizás no sea necesariamente un requisito tener una curva con velocidad constante.

Slerp produce la trayectoria de interpolación más corta posible entre dos cuaternios sobre la esfera unitaria, lo que redundaría en que su velocidad angular se manifieste constante. Esto hace que se pueda afirmar que Slerp es la curva de interpolación óptima entre dos rotaciones.

Aunque no está incluida la opción Squad en el desarrollo de la aplicación, es importante indicar que esta entrega una muy buena curva de interpolación, siendo necesario especificar que cuando se hace referencia a “muy buena curva” se intenta describir como continua y diferenciable, esto a pesar de que hay que reconocer que incluso esta comparación puede ser cualitativamente incompleta.

En definitiva, usar ángulos de Euler representados por matrices lleva a una serie de problemas, siendo el más frecuente, la presencia de gimbal lock; la representación con cuaternios posee una interpretación geométrica más natural y una parametrización de la rotación que no depende del sistema de coordenadas. La única ventaja real de las matrices es la posibilidad de representar todas las demás transformaciones. (Dam, Koch, Lillholm, 1998).

De todas las alternativas analizadas en este trabajo e incluidas en la aplicación desarrollada en MatLab, Slerp ofrece la mejor opción para la representación de las rotaciones.

FIGURA 12
Análisis de trayectorias y su simulación en Matlab utilizando diversas técnicas de interpolación lineal
Captura de pantalla completa de la aplicación



REFERENCIAS BIBLIOGRÁFICAS Y ELECTRÓNICAS

- [1]. **DAM, E. KOCH, M. LILLHOLM, M.** (1998). “*Quaternions, Interpolation and Animation*”. Technical Report DIKU-TR-98/5. University of Copenhagen. Denmark.
- [2]. **EULER, L.** (1752). “*Decouverte d'un nouveau principe de mécanique*” Original mente de “*Mémoires de l'académie des sciences de Berlin 6*”, 1752, pp. 185-217.
- [3]. **FOLEY et al** (1995). James D. Foley, Andries van Dam, Steven K. Feiner & John F. Hughes. “*Computer Graphics Principles and Practice*”. Reading, Massachusetts.
- [4]. **HAMILTON, W.** (1853). “*Lectures on quaternions*”. Hodges and Smith & Co., Dublin, 1853.
- [5]. **KIM et al** (1996) Myoung-Jun Kim, Myoung-Soo Kim & Sung Yong Shin. “*A compact differential for the first derivative of a unit quaternion curve*”. The Journal of Visualization and Computer Animation.
- [6]. **SHOEMAKE, K.** (1987). “*Quaternions calculus and fast animations*”. Siggraph course notes.
- [7]. **WATT&WATT** (1992). Alan Watt & Mark Watt. “*Advance Animation and Rendering Techniques theory and practice*”. Chapter 15. Adisson Wesley, Wokingham, England 1992.