

Laboratorio Spline Motion Control en la Grúa Maya UN

Andrés Holguín Restrepo
Ingeniería Mecatrónica
Universidad Nacional de Colombia
Bogotá, Colombia
aholguinr@unal.edu.co

Resumen—En este informe de laboratorio se va a realizar un ensayo de trayectoria tipo spline de 30 puntos tridimensional enfocado para su implementación en la Grúa Maya UN. A partir de un apuntador láser se logra generar una proyección en el plano x-y de la posición actual, lo que permite generar una trazabilidad de los resultados obtenidos y de este modo lograr realizar un análisis de resultados prácticos y teóricos con el fin de verificar la precisión y exactitud de la grúa Maya UN. De este modo, se logra caracterizar los resultados obtenidos de una forma clara, exponiendo las limitaciones que se tenían y las posibles mejoras que se podrán implementar en futuras ocasiones.

I. TRAYECTORIA SPLINE

Inicialmente, se va a definir una trayectoria tridimensional dada por las siguientes ecuaciones:

$$t \in [0 \ 2\pi] \quad (1)$$

$$x(t) = 1 + 0.5 \sin(t) e^{\frac{-t}{2\pi}} \quad (2)$$

$$y(t) = 1 + 0.5 \cos(t) e^{\frac{-t}{2\pi}} \quad (3)$$

$$z(t) = 1.5 - \frac{t}{2\pi} \quad (4)$$

En la figura 1 se evidencia para este rango de t los valores de x , y y z .

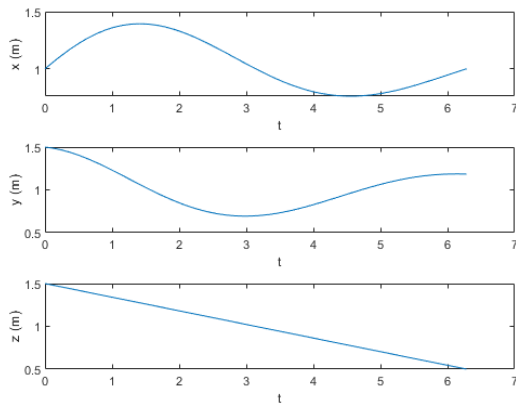


Figura 1. Trayectoria inicial de movimiento

De esta trayectoria, solo se van a utilizar nueve puntos iniciales para realizar posteriormente los procesos de interpolación, los cuales se pueden ver en la tabla I.

Tabla I
VALORES INICIALES DE TRAYECTORIA

N	$x(m)$	$y(m)$	$z(m)$
1	1.0000	1.5000	1.5000
2	1.3120	1.3120	1.3750
3	1.3894	1.0000	1.2500
4	1.2430	0.7570	1.1250
5	1.0000	0.6967	1.0000
6	0.8108	0.8108	0.8750
7	0.7638	1.0000	0.7500
8	0.8526	1.1474	0.6250
9	1.0000	1.1839	0.5000

A partir de estos nueve valores, se realiza una interpolación spline de un total de 30 puntos, los cuales se van a utilizar para recrear la trayectoria original. Véase la tabla II para identificar esta totalidad de puntos.

Ahora bien, con base en estos 30 puntos, es posible realizar graficar la trayectoria mediante interpolación spline ya sea en un marco tridimensional, o en una proyección 2D en el plano $x-y$. Véase las figuras 2 y 3 para evidenciar esta trayectoria respectivamente. A partir de estas trayectorias, es posible

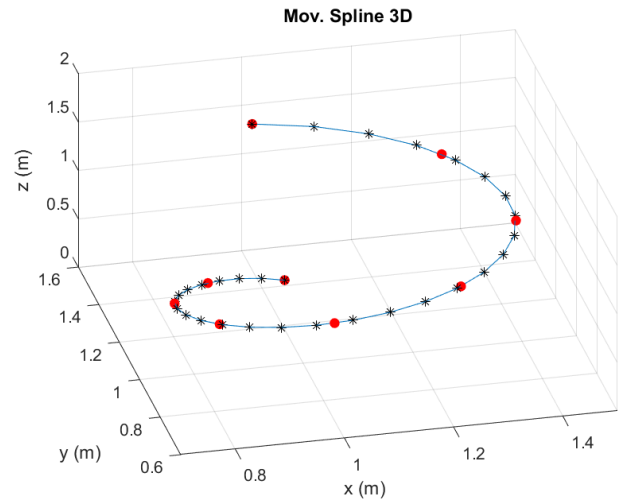


Figura 2. Trayectoria spline 3D

calcular el error teórico que se genera entre la trayectoria tipo spline y el valor real esperado mediante la parametrización de la trayectoria. Estos errores se pueden ver mediante sus

Tabla II
30 PUNTOS DE TRAYECTORIA POR INTEROPLACIÓN SPLINE

N	x (m)	y (m)	z (m)
1	1.00	1.50	1.50
2	1.11	1.47	1.47
3	1.20	1.42	1.43
4	1.27	1.36	1.40
5	1.33	1.28	1.36
6	1.37	1.20	1.33
7	1.39	1.11	1.29
8	1.39	1.02	1.26
9	1.37	0.94	1.22
10	1.34	0.86	1.19
11	1.29	0.80	1.16
12	1.24	0.75	1.12
13	1.17	0.72	1.09
14	1.10	0.70	1.05
15	1.03	0.69	1.02
16	0.97	0.70	0.98
17	0.91	0.73	0.95
18	0.86	0.76	0.91
19	0.82	0.80	0.88
20	0.79	0.85	0.84
21	0.77	0.91	0.81
22	0.76	0.96	0.78
23	0.77	1.01	0.74
24	0.78	1.06	0.71
25	0.81	1.10	0.67
26	0.84	1.14	0.64
27	0.88	1.16	0.60
28	0.92	1.18	0.57
29	0.96	1.19	0.53
30	1.00	1.18	0.50

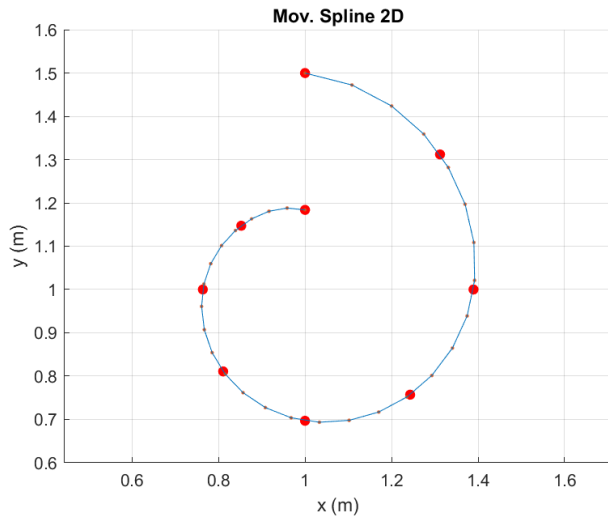


Figura 3. Trayectoria spline 2D

componentes en la figura 4 o por la norma euclidiana del error en la figura 5.

Hecha esta observación, es posible pasar a la etapa de implementación de la Grúa Maya UN.

II. GRÚA MAYA

La grúa maya es un manipulador híbrido serie-paralelo. Tiene tres actuadores principales que permiten llevar el TCP a cualquier punto dentro de su espacio de trabajo, y un gripper

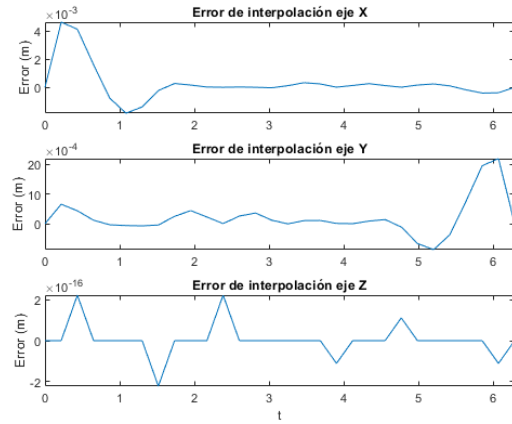


Figura 4. Error por componentes de trayectoria Spline

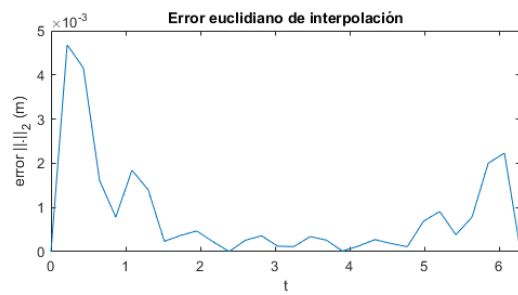


Figura 5. Error por norma euclidiana de trayectoria Spline

neumático que permite realizar el agarre de piezas de trabajo. En las figuras 6 y 7 puede ver un esquemático y una foto real de este Robot respectivamente.

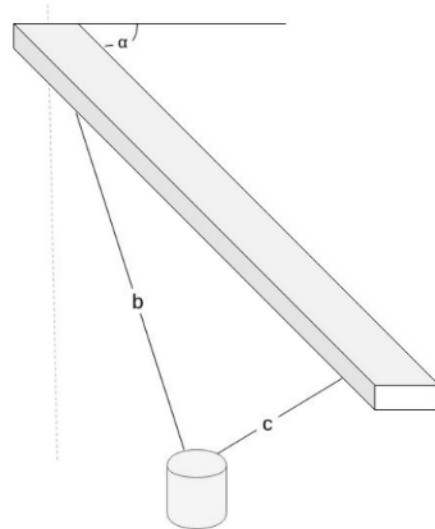


Figura 6. Error por componentes de trayectoria Spline

Así mismo, los tres actuadores implementados tienen funciones diferenciales. El primer motor genera un giro directo con el ángulo α visto en la figura 6. El segundo motor controla



Figura 7. Grúa Maya UN

la extensión de la correa síncrona en el segmento b , y el tercer motor, de manera similar, controla la extensión de la correa en el segmento c .

II-A. Análisis cinemático

Para realizar acordemente un análisis cinemático de la Grúa Maya, lo ideal es realizar una transformación del sistema de coordenadas cartesianas a un sistema de coordenadas cilíndricas, el cual permite simplificar los valores de los actuadores correspondientes.

$$\theta_1 = \text{atan2}(y, x) \quad (5)$$

$$\rho = \sqrt{x^2 + y^2} \quad (6)$$

$$z = z \quad (7)$$

En la figura 8 se puede evidenciar una asignación de parámetros a los valores de la grúa.

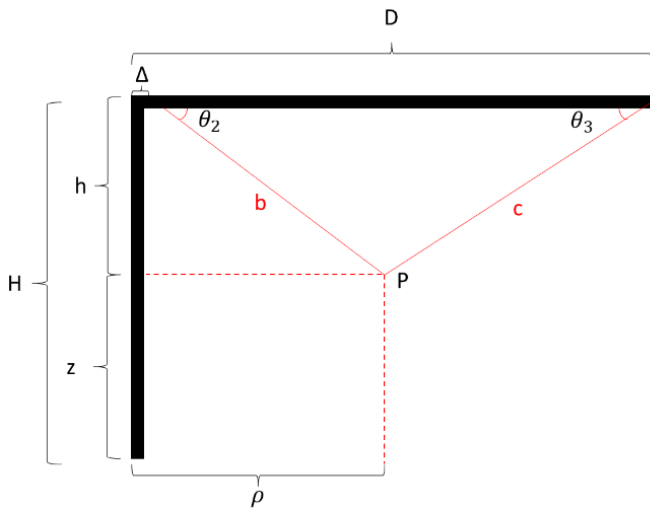


Figura 8. Parametrización desde vista lateral de Grúa Maya UN

Al realizar un análisis de los parámetros, se puede determinar de manera sencilla mediante el método geométrico, al cinemática directa e inversa de este manipulador, donde se tiene en cuenta el marco de coordenadas cilíndricas. De

este modo, se encuentran las ecuaciones de cinemática inversa como:

$$\theta_1 = \text{atan2}(y, x) \quad (8)$$

$$b = \sqrt{(\Delta - \rho)^2 + (H - z)^2} \quad (9)$$

$$c = \sqrt{(D - \rho)^2 + (H - z)^2} \quad (10)$$

A partir de estas ecuaciones, es posible generar las condiciones necesarias de los actuadores para generar las trayectorias deseadas. En los anexos de esta entrega se evidencian videos de simulación en Matlab sobre este sistema.

Llegado a este punto, es posible pasar al software de manipulación de la Grúa Maya.

II-B. Software

Se deben hacer varias aclaraciones sobre el Software de uso diseñado por estudiantes pasados del curso de APM.

- **Ubuntu:** Este proyecto fue escrito para Ros Kinetic y requiere Ubuntu Xenial (16.04). Este documento asume que el usuario utilizará este sistema operativo, por lo que cada comando está escrito para él. Se requiere cierta familiaridad con la terminal de comandos.
- **Git:** Cada fragmento de código mencionado en este artículo se encuentra en un repositorio de git. Los estudiantes deben familiarizarse con esta herramienta de control de versiones, ya que se mencionará en varias secciones siguientes. Será necesario tener un usuario en gitlab para descargar el código y realizar contribuciones. Gitlab es la plataforma que actualmente alberga el software de LabFabEx.
- **Galiltools:** Galil tools es un software que permite la comunicación directa con el controlador Galil. El estudiante que desee contribuir a este proyecto debe familiarizarse con la interfaz gráfica de usuario y la funcionalidad de esta herramienta, ya que sirve como canal de prueba para los comandos, así como una interfaz de respaldo para el robot.
- **ROS Kinetic:** ROS es el marco de trabajo en el que se ha desarrollado este proyecto. Utiliza herramientas de construcción y comandos del marco de trabajo, así como varias pautas para su estructura de archivos y estilo de codificación. Todos los datos transmitidos en el sistema se realizan a través de mensajes en tópicos y servicios, por lo que el estudiante debe estar familiarizado con los principios básicos y las funcionalidades del marco de trabajo.
- **Python:** Python es el lenguaje en el que se ha escrito la mayor parte del código base. Todos los módulos, excepto ros-coms, están escritos en Python 2.7.12 (ROS no admite oficialmente Python 3).
- **NodeJS / Yarn / npm:** Se utiliza NodeJS para el protocolo PRIA. Si se desea la comunicación con Firebase, se debe instalar NodeJS. NVM puede utilizarse para gestionar las versiones de NodeJS. El módulo ros-coms se ha probado con la última versión LST, node v10.15.3 (npm v6.4.1). Aunque Yarn es el gestor de paquetes

preferido en este proyecto, también se puede utilizar npm. Los comandos para ambos se proporcionarán cuando sea necesario.

- **Typescript:** El módulo ros-coms está escrito en TypeScript. Aunque contribuir a ros-coms está fuera del alcance de este documento, puede ser útil comprender cómo el paquete `maya_process` recibe la información del proceso.

A partir de esto, si se desea usar el Software en modo simulación, se debe ejecutar el comando `roslaunch maya_process process.launch`. Si se desea implementar en la vida real, se utiliza el comando `roslaunch maya_process process.launch sim_mode:=false`. De este modo, se puede operar ya sea en físico como en simulación.

En la figura 9 se puede evidenciar la estructura de la GUI para la manipulación del software de la Grúa Maya UN.

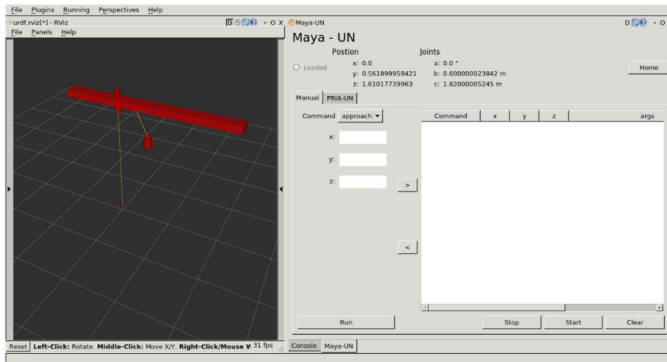


Figura 9. GUI de implementación de Grúa Maya UN

III. PROBLEMAS PREVIOS A LA IMPLEMENTACIÓN

Antes de realizar la implementación final, en la etapa de pruebas de la Grúa Maya UN, se determinan diversos errores que debieron ser solucionados y ser tenidos en cuenta para la implementación definitiva. A continuación se hace mención de ellas:

- **Operación de Home:** Al realizar pruebas de funcionamiento, se identifican problemas con la calibración del parámetro c de la Grúa durante la operación de Home. De este modo, como no se lograba establecer una referencia adecuada de este parámetro, después del primer movimiento, la Grúa Maya UN perdía la referencia de coordenadas y no podía seguir continuando.
- **Capacidad de Puntos:** El software de implementación de la Grúa Maya UN aparenta tener un problema en el bus de datos de comunicaciones con el controlador Galil del robot. Al realizar pruebas de simulación, es posible enviar una gran cantidad de puntos en la trayectoria y de este modo tener una trayectoria más completa, en este caso siendo una trayectoria de 30 puntos. Sin embargo, en la implementación física, no es posible enviar tantos puntos, ya que se genera un error desconocido que hace que la rutina deba detenerse y provocar un error. La solución a esto fue enviar subrutinas de la trayectoria en baches de seis puntos, por lo que implica que en cinco

ocasiones se debía agregar los puntos manualmente para continuar con la trayectoria.

- **Análisis de spline:** Debido a que se está operando bajo un movimiento tridimensional en la Grúa Maya UN, se dificulta poder realizar un análisis real de la precisión y exactitud de los resultados obtenidos. Por esta razón, la manera de arreglar esta problemática fue utilizar el gripper de la Grúa para agarrar un laser SD303 en funcionamiento continuo para generar una proyección en el plano $x - y$, así mismo, para determinar las coordenadas puntuales de este puntero láser, se imprime en un formato A0, una cuadrícula escala real de rangos $x \in [70, 140]cm$ y $y \in [60, 150]$ y se calibra el punto $[100, 100]cm$, donde de este punto se genera el inicio del recorrido de la trayectoria.

IV. IMPLEMENTACIÓN

Ahora, se va a pasar a las características de comportamiento que se tuvo en la implementación. Debido a la rigidez dinámica del sistema, y que la herramienta es sostenida por dos correas, siendo estas los eslabones b y c del manipulador, se tienen oscilaciones bastante prominentes y prolongadas al realizar movimientos de alto desplazamiento en el plano $x - y$, ya que esto implica un movimiento intrínseco de los tres actuadores. Además, cabe mencionar la falta de optimización del controlador del actuador 1, ya que se evidencia un comportamiento bastante subamortiguado para llegar a la referencia establecida. Todo esto implica que para analizar el resultado del movimiento de la Grúa Maya en estado estable, es necesario esperar a que se cumpla el tiempo del estado transitorio, el cual estaba rondando los 15-20 segundos para que las oscilaciones del gripper se detuvieran.

Sumado a esto, otro detalle importante de la implementación es que solo se obtienen valores experimentales en la proyección del plano $x - y$ mediante el laser SD303. Sin embargo, debido a problemas de ajuste, fue necesario generar un soporte de tela para que el gripper pudiera sujetar firmemente el láser. Al realizar este montaje, se generaban errores de giro del láser, donde este dejaba de tener dirección normal al suelo, lo cual generaba errores en la proyección en el suelo. Cabe resaltar que este error se minimiza al calibrar el láser sobre el punto $[100, 100]cm$, aunque se reconoce que este error debe ser contemplado para análisis más profundos de resultados.

Debido a la limitante de comunicación

V. RESULTADOS

Con base en la implementación realizada, fue posible realizar pruebas de exactitud y precisión dadas en los resultados obtenidos. De este modo, se realizan diversas pruebas para comprobar la precisión de la Grúa, la cual se puede garantizar siempre y cuando la orientación y el ángulo de orientación del puntero láser se mantengan uniformes durante las pruebas. Véase la figura 10 De este modo, en la tabla III se evidencia el promedio de los resultados experimentales obtenidos para los 30 puntos de la trayectoria. A partir de estos valores, es posible graficar estos resultados versus los valores teóricos

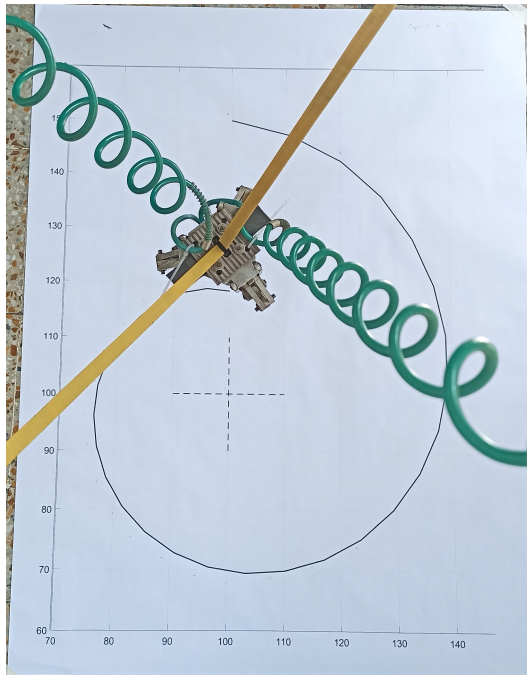


Figura 10. Montaje experimental

mencionados anteriormente para corroborar la concordancia entre estas dos pruebas, esto visto en la figura 11. Ya teniendo

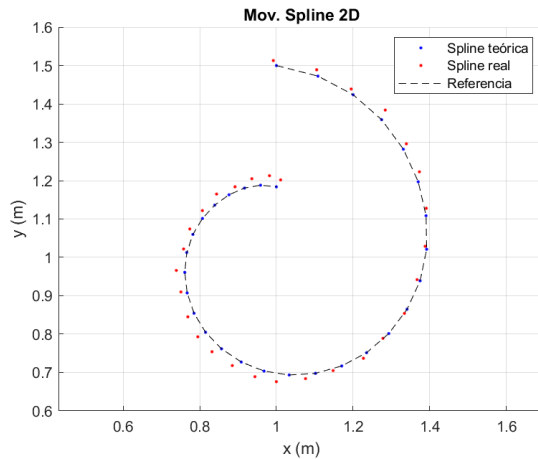


Figura 11. Trayectoria spline real vs teórica

una primera vista de comparación de los resultados, se puede realizar el cálculo de error de trayectoria real generado en la implementación de la Grúa Maya UN versus el calculado teóricamente en secciones anteriores, visto en la figura 12.

Como se puede evidenciar, y como era de esperarse con base en observaciones anteriores, el error generado en la implementación es considerablemente mayor, donde al realizar el cálculo de la relación promedio de error, se tiene un error promedio de implementación de 300 veces mayor que el teórico, teniendo en cuenta únicamente las componentes proyectadas sobre el plano $x - y$.

Tabla III
VALORES DE IMPLEMENTACIÓN REAL DE SPLINE DE 30 PUNTOS.

N	x (cm)	y (cm)
1	99.2	151.3
2	110.5	148.9
3	119.5	143.9
4	128.4	138.4
5	133.9	129.6
6	137.3	122.3
7	139.1	112.8
8	138.8	102.9
9	136.7	94.2
10	133.4	85.4
11	127.8	78.9
12	122.7	73.7
13	114.8	70.5
14	107.6	68.4
15	100.0	67.6
16	94.4	68.9
17	88.5	71.8
18	83.2	75.4
19	79.5	79.3
20	76.9	84.5
21	75.1	91.0
22	73.9	96.6
23	75.8	102.2
24	77.4	107.4
25	80.7	112.2
26	84.4	116.5
27	89.2	118.4
28	93.6	120.5
29	98.2	121.3
30	101.1	120.2

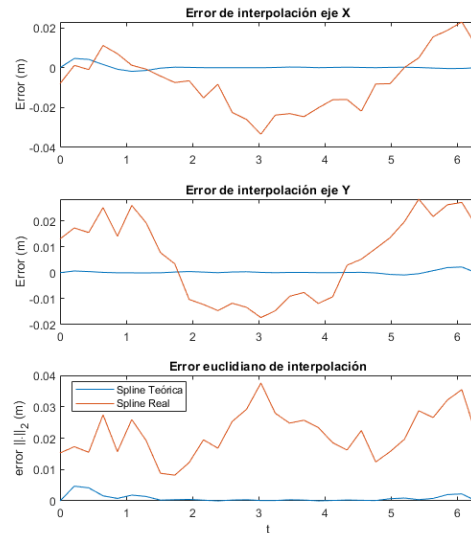


Figura 12. Valores de error de trayectoria spline real vs teórico

VI. CONCLUSIONES Y RECOMENDACIONES

La Grúa Maya UN es un manipulador capaz de generar movimientos tridimensionales controlados mediante trayectorias tipo spline lo cual le permite generar diversas tareas de automatización y de robótica colaborativa. Sin embargo, es necesario tener en cuenta los problemas de rigidez dinámica

del sistema y de las oscilaciones generadas por esta y por los parámetros de los controladores de los actuadores.

Se recomienda para futuros trabajos la optimización de los parámetros del controlador del actuador que controla el ángulo α , ya que se observa que es posible ajustarlo para obtener un comportamiento sobreamortiguado y así reducir las oscilaciones físicas del sistema.

Además, se recomienda una actualización del software de implementación del robot, ya que en la implementación, la comunicación con el controlador era bastante inestable, lo cual limitaba la posibilidad de enviar altas cantidades de puntos simultáneos para generar una trayectoria continua, siendo esta cantidad nunca superior a 10 puntos y a veces siendo limitada a seis valores por rutina.

REFERENCIAS

[1] M. S. Paing and N. Uchiyama, "Spline-Based Time-Optimal Control for Smooth Trajectory Generation of CNC Machines with Geometric Constraints, in IEEE Transactions on Automation Science and Engineering, vol. 18, no. 1, pp. 171-182, Jan. 2021, doi: 10.1109/TASE.2020.3008631.

[2] S. Araujo and E. Córdoba Nieto, "Maya Robot Maya-ROS-UN," Technical Manual, LabFabEx - Dima UN, Departamento de Ingeniería Mecánica Mecatrónica, Universidad Nacional de Colombia, Sede Bogotá, May 2019.