ISSN: 1692-7257 - Volumen 2 – Número 22 - 2013

Revista Colombiana de Tecnologías de Avanzada

Recibido: 03 de agosto de 2012
Aceptado: 19 de octubre de 2012

FUZZY SPEED CONTROL FOR TRAJECTORY FOLLOW-UP OF A ROBOT MOVING ON VERTICAL SURFACES

CONTROL FUZZY DE VELOCIDAD EN UN ROBOT MÓVIL PARA EL SEGUIMIENTO DE TRAYECTORIAS EN EL DESPLAZAMIENTO DE SUPERFICIES VERTICALES

Ing. (c) Roger F. Castro, Ing. (c) José Nicolás Moreno M
MSc. Hernán González Acuña, PhD. Omar Lengerke Pérez

Universidad Autónoma de Bucaramanga - UNAB.
Programa de Ingeniería Mecatrónica, Grupo de Investigación en Control y Mecatrónica.
Avenida 42 No. 48 – 11, Bucaramanga, Santander, Colombia.
Tel.: +(57) (7) 643 6111, Ext. 481.
E-mail: {rcastro760, jmoreno4, hgonzalez3, olengerke}@unab.edu.co

Abstract: This paper shows the design of fuzzy speed control of trajectories in a mobile robot. This robot can move on vertical surfaces of storage tanks for exploration, inspection and cleaning tasks. While the first step involvedactuators (DC motors) identification, the second step presented fuzzy control speed, which can be used for trajectory control. Finally, the genetic algorithm was implemented to obtain optimal trajectory of critical points on the surface where the robot is to follow its predetermined trajectory.

Keywords: Fuzzy control, mobile robot, climbing robot, trajectories control.

Resumen: Este artículo presenta el diseño de un controlador fuzzy aplicado en el control de trayectorias en un robot móvil que se desplazara sobre superficies verticales con la finalidad de realizar operaciones de inspección y limpieza en tanques de almacenamiento de hidrocarburos. En la primera etapa se realiza la identificación de cada uno de los actuadores (motores DC). En la segunda etapa se presenta el diseño del controlador de velocidad fuzzy con el cual se podrán seguir las trayectorias determinadas. Y finalmente se presenta el algoritmo que le permitirá al robot determinar cuáles son los puntos más críticos de la superficie a inspeccionar y cual será camino que deberá seguir el robot móvil sobre la superficie.

Palabras clave: Control fuzzy, robot móvil, trayectorias, superficies verticales.

1. INTRODUCCIÓN

Una de las principales funciones de la robótica es brindar soluciones a las diferentes tareas que puedan representar gran complejidad para el ser humano. Una de estas tareas es la inspección y limpieza de superficies verticales como tanques de almacenamiento de hidrocarburos o grandes edificios.

Algunas de estas aplicaciones son: Nagakubo y Hirose, 1994, realizaron uno de los primero robots que podían escalar superficies verticales, este robot fue desarrollado en el instituto de tecnología Tokiopara limpieza en edificaciones y puentes, tiene un peso de 45 Kg y una velocidad máxima de 7,4 m/min. Zhang *et al*., 2007, se presenta un robot neumático que puede escalar edificios de 50 m de altura.

Longo et al., 2005, presenta el diseño del robot llamado Alicia fue construido principalmente para la inspección de superficies verticales no porosas.

Gonzalez et al., 2012 presenta el diseño de un robot para la limpieza de superficies verticales en tanques de almacenamiento de hidrocarburos.

2. IDENTIFICACIÓN DEL SISTEMA

Para la identificación de un sistema se realizo mediante el uso de modelos paramétricos como ARX, ARMAX, OE y BJ, (Garrido y Moreno, 2003), que consisten en tomar la señal de respuesta transitoria del sistema ante voltajes de entrada en una zona lineal, de esta manera tratar de obtener el modelo matemático que mejor represente las características dinámicas del modelo real.

La Tabla 1 presenta los datos obtenidos para identificar el rango de funcionamiento del motor, aquí se determina con que voltaje el motor comienza su movimiento, esto es importante para no generar señales en la zona muerta del motor.

Tabla 1: Rango de funcionamiento del motor

Voltaje Entrada	Velocidad Angular
0,055	0
0,111	0
0,166	0
0,222	2,9
0,277	4,5
0,333	6,4
0,388	7,8
0,444	10,7
0,5	12,89
0,555	15,4
0,611	17,7

En la Fig. 1 se presenta la relación voltaje de entrada vs velocidad de salida del motor, se observa que la respuesta es mas lineal entre el rango de 0,4 V y 0,65 V de entrada se opta por elegir entre límite inferior 0,4 V y limite superior 0,5V.

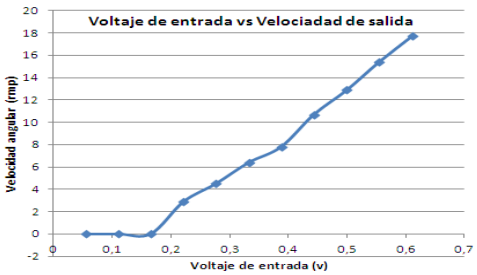

Fig. 1. Relación de voltaje de entrada velocidad de salida del motor DC.

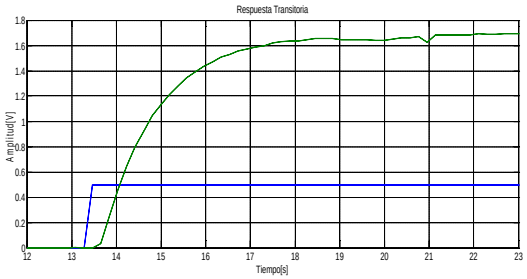

Fig. 2. Respuesta transitoria a una entrada paso de 0.5 v.

2.1 Identificación con modelos paramétricos.

La identificación del sistema se realizo con los datos adquiridos de aplicar 2 diferentes señales PRBS las cuales son presentas en la las cuales se toman lasmuestras de la respuesta transitoria ante voltajes de entrada encontrados en la zona lineal de esta manera tratar de obtener las posibles variaciones presentes en la planta y así poder obtener un modelo matemático más fiel y que cumpla con las características dinámicas del sistema.

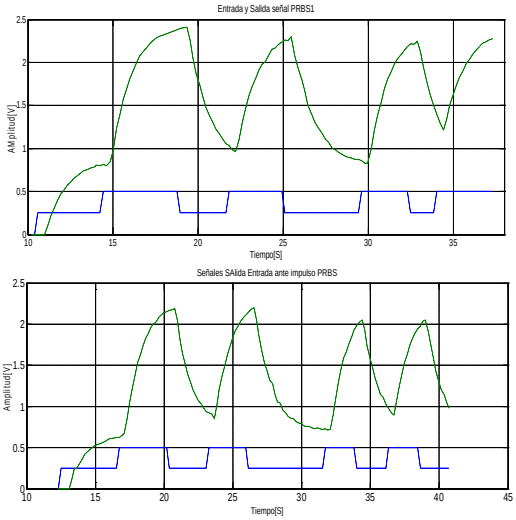

Fig. 3. Señales PRBS aplicadas y respuesta del sistema.

Usando la *Toolbox* de identificación de Matlab y con ayuda de los modelos paramétricos se buscan los modelos matemáticos con los mejores *bestfits* como se observa en la Fig. 4. El *bestfit* representa el porcentaje de similitud entre la señal real y la señal de los modelos adquiridos, sin embargo es necesario analizar la respuesta de cada uno de estos modelos y determinar que las respuestas son lógicas y similares con la señal real. Por esta razón fue escogido el oe232 que presenta un 76.25%.

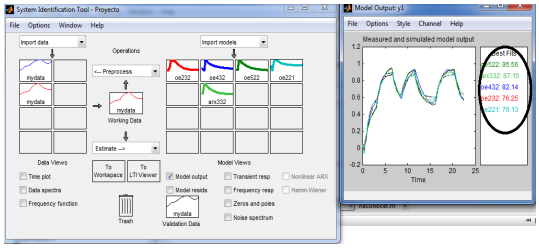

Fig. 4. Modelos de identificación obtenidos.

En la Fig. 5 se presenta la respuesta transitoria del modelo oe232 ante una entrada escalón, este modelo presenta una mayor similitud con la respuesta real.

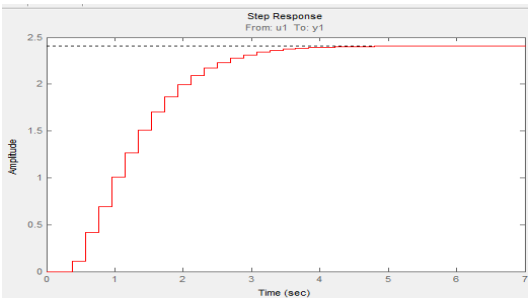

Fig. 5. Modelos de identificación obtenidos.

En la ecuación (1) se presenta la ecuación matemática del modelo oe221:

$$G(z)=\frac{0.1073z+0.2254}{z^3-0.7467z^2-0.4806z+0.3656} \qquad (1)$$

Con un periodo de muestreo de 192 mseg limitado por la adquisición del hardware Labview.

3. DISEÑO CONTROLADOR FUZZY

El diseño preliminar del controlador fuzzy se realizó con la configuración de controlador proporcional en el cual la salida tenía una relación directa con el error como se presenta en la *Fig. 6.*

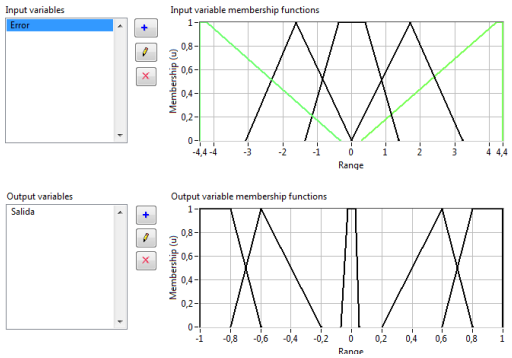

Fig. 6. Conjuntos iniciales del controlador fuzzy.

La Fig. 7 se presenta la respuesta del controlador de velocidad diseñado inicialmente, se observa en la figura que la señal de respuesta tiene un tiempo establecimiento demasiado grande lo cual no lo hace un controlador viable para ser usado. Por esta razón se debió sintonizar mejor el controlador de velocidad.

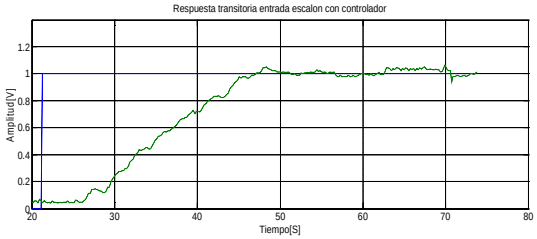

Fig. 7. Respuesta inicial del controlador de velocidad.

Finalmente el controlador fuzzy diseñado en Matlab se define 3 grupos esenciales, 2 entradas y una salida. La primera entrada es la señal de error,Fig. 8, se tienen en cuenta errores positivos correspondientes a *setpoints* mayores a la señal de velocidad, un error *zero* donde se integra un rango en el cual el error es permisible y por ultimo errores negativos que denotan que la señal de medida de velocidad es mayor al punto de consigna determinado por el usuario.

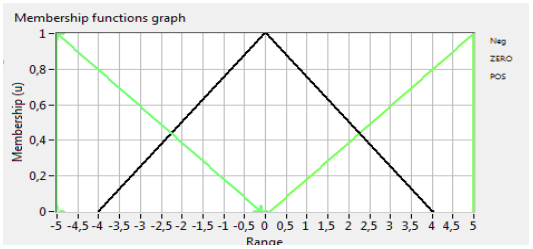

Fig. 8. Entrada del error.

La segunda entrada, Fig. 9, es la derivada del error encargado de entregarle velocidad al sistema y definir la variación del error y por tanto la eficiencia de la acción entregada en cada momento.

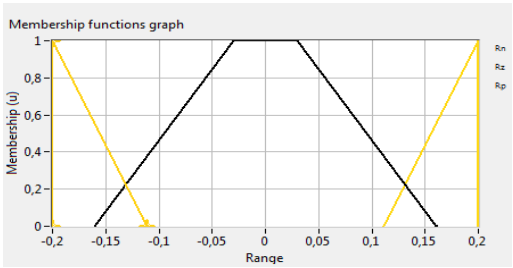

Fig. 9. Derivada del error.

La acción de salida, Fig. 10, está definida en el siguiente grupo y fue aplicada gracias a pruebas en la simulación que hicieran posible la estabilidad del sistema ante la presencia del controlador.

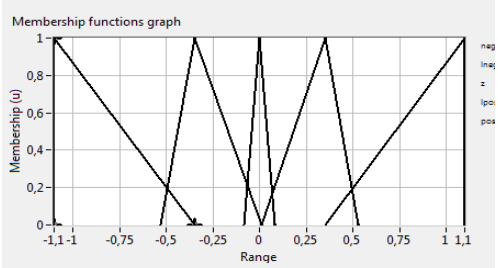

Fig. 10. Conjunto de salida.

En la Fig. 11, se presentan las reglas que fueron programadas en el controlador fuzzy.

1. IF 'Error' IS 'Neg' AND 'RATE' IS 'NEG' THEN 'Kd' IS 'Pos'
2. IF 'Error' IS 'Neg' AND 'RATE' IS 'MED' THEN 'Kd' IS 'z'
3. IF 'Error' IS 'Neg' AND 'RATE' IS 'ALTO' THEN 'Kd' IS 'z'
4. IF 'Error' IS 'ZERO' AND 'RATE' IS 'NEG' THEN 'Kd' IS 'Ipos'
5. IF 'Error' IS 'ZERO' AND 'RATE' IS 'MED' THEN 'Kd' IS 'z'
6. IF 'Error' IS 'ZERO' AND 'RATE' IS 'ALTO' THEN 'Kd' IS 'lneg'
7. IF 'Error' IS 'POS' AND 'RATE' IS 'NEG' THEN 'Kd' IS 'z'
8. IF 'Error' IS 'POS' AND 'RATE' IS 'MED' THEN 'Kd' IS 'Ipos'
9. IF 'Error' IS 'POS' AND 'RATE' IS 'ALTO' THEN 'Kd' IS 'Pos'

Fig. 11. Reglas del controlador Fuzzy.

Tabla 2: Base de reglas

	NG	ME	Al
NG	PO	Z	Z
Ze	lneg	Z	lneg
PO	Z	Lp	PO

La Fig. 12 presenta el lazo de control diseñado para realizar el control de velocidad en la tiene la misma señal de entrada para el sistema real y para el controlador diseñado.

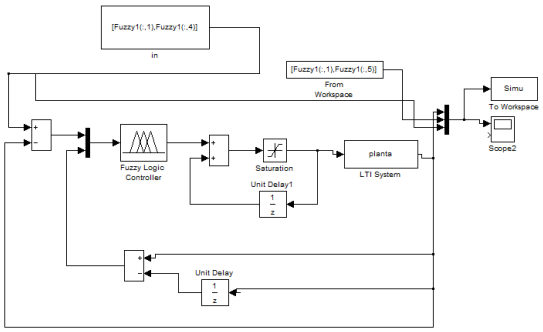

Fig. 12. Lazo de control diseñado para el control de velocidad.

Una vez sintonizado el controlador fuzzy se realizo una comparación entre la respuesta real y la respuesta obtenida en la simulación. Esta comparación es presentada en la Fig. 13, donde se uso la misma entrada para las 2 señales y la misma escala de tiempo. Al comparar las graficas encontramos ciertas diferencias como el tiempo de establecimiento de 1,23 s más lento para alcanzar el punto de consigna y un sobrepaso del 9 %, sin embargo se obtiene un seguimiento aceptable de la señal después de la sintonización del controlador.

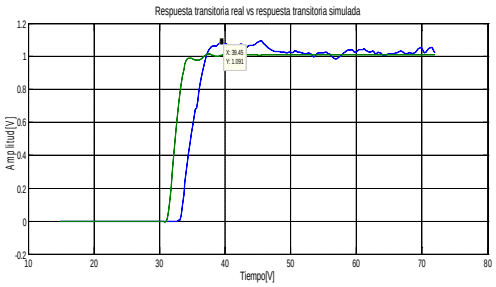

Fig. 13. Comparación respuesta real y la respuesta simulada

En la Fig. 14 se presenta la grafica de contorno de las reglas fuzzy establecidas para el correcto funcionamiento del controlador.

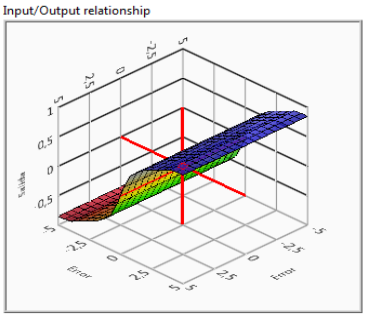

Fig. 14. Gráfica de contorno del controlador de velocidad.

En la Fig. 15 se evaluó la respuesta del controlador sobre la oruga derecha, de esta forma se determina si su respuesta es aceptable a partir de la necesidady decidir si es usado o sintonizado nuevamente. Se observa que así como en la simulación el sistema tiene una buena respuesta y se procede a implementar en la oruga derecha e izquierda del robot móvil.

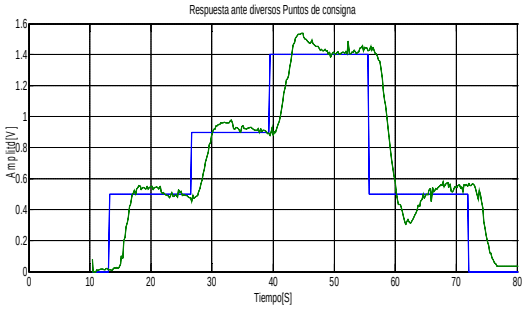

Fig. 15. Respuesta de rueda derecha a la entrada.

4.1 Resultados

El controlador diseñado para cada una de las orugas se prueba con una señal de entrad paso donde se aplican diferentes *setpoints* en el tiempo.

En la Fig. 16, se presenta la señal de respuesta del control de velocidad sobre la rueda derecha.

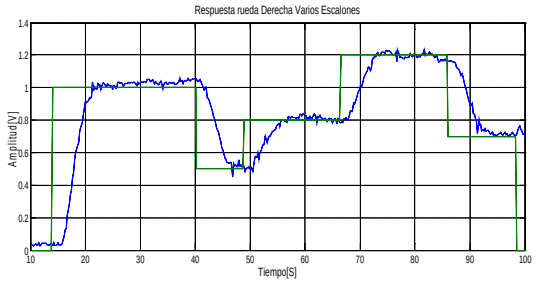

Fig. 16. Respuesta de rueda derecha a la entrada.

En la Fig. 17, se presenta la señal de respuesta del control de velocidad sobre la rueda izquierda.

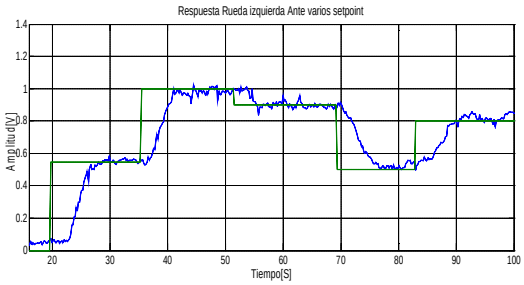

Fig. 17. Respuesta de rueda izquierda a la entrada.

4. ROBOT MÓVIL

4.2 Análisis estático del robot móvil.

Equilibrio de fuerzas en el eje X, (2) and Y, (3):

$$\sum F_x = 0$$
$$F_{adh} = R_{w1} + R_{w2} + R_{w3} + R_{w4} \quad (2)$$

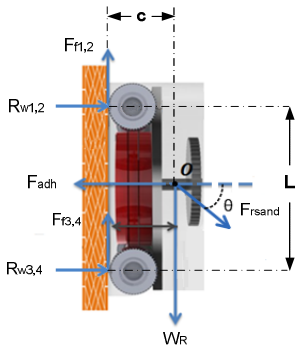

Fig. 18. Diagrama de cuerpo libre.

$$\sum F_y = 0$$
$$W_r = F_{f1} + F_{f2} + F_{f3} + F_{f4} \quad (3)$$

Debido a la simetríadel robot las fuerzas en R_{w3} y R_{w4} son iguales, Se realiza la sumatoria de torques alrededor del punto F obteniendo:

$$\sum T_F = 0$$
$$R_{w3,4} \cdot L = F_{adh} \cdot \frac{L}{2} + W_R \cdot c \quad (4)$$

La fuerza de adhesión es obtenida de la ecuación (4) cuando el robot está en la superficie vertical. La fuerza de adhesión, (5), es función del peso del robot móvil y las fuerzas de rozamiento de las orugas con la pared.

$$F_{adh} = \frac{2}{L}\left(R_{w3,4} \cdot L - W_R \cdot c\right) \quad (5)$$

4.3 Modelo cinemática del robot móvil

Un robot diferencial, Fig. 19, posee una menor restricción en sus movimientos, su movimiento depende directamente de la diferencia de velocidad entre su rueda izquierda y derecha.

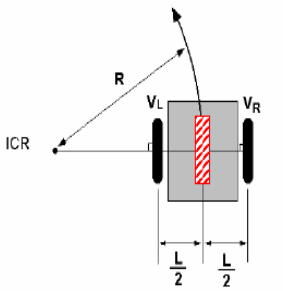

Fig. 19. Robot de configuración diferencial.

En la ecuación (6) se presenta al modelo cinemático que rige el comportamiento del robot.

$$v = \frac{V_L + V_R}{2} = \frac{R\left(w_L + w_R\right)}{2}$$
$$w = \frac{V_L + V_R}{2} = \frac{R\left(w_L + w_R\right)}{2} \quad (6)$$

De la ecuación (6) se calcula la posición y orientación del robot mediante la integración de su velocidad.

$$x(t) = x(t_0) + \int v(t)\cos\left[q(t)\right]dt$$
$$y(t) = y(t_0) + \int v(t)\sin\left[q(t)\right]dt \quad (7)$$
$$q(t) = q(t_0) + \int w(t)\,dt$$

5. DISEÑO EN CAD DE ROBOT MÓVIL CON LOCOMOCIÓN DIFERENCIAL

El robot diseñado inicialmente para cumplir con la tarea de exploración de superficies verticales es presentado en la Fig. 20, González *et al.*, 2012.

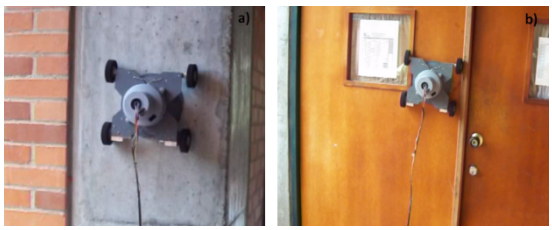

Fig. 20. Diseño preliminar robot móvil.

Para reducir los problemas de deslizamiento se fue diseñado un robot móvil con tracción por orugas esta le permitirá reducir la probabilidad de deslizamiento en cada una de las mismas ay que se tendrá mayor superficie de contacto con la superficie, Fig. 21.

Fig. 21. Diseño en CAD de robot con oruga

En la Fig. 22se presenta el robot finalizado con el cual se realizara la exploración de las superficies verticales.

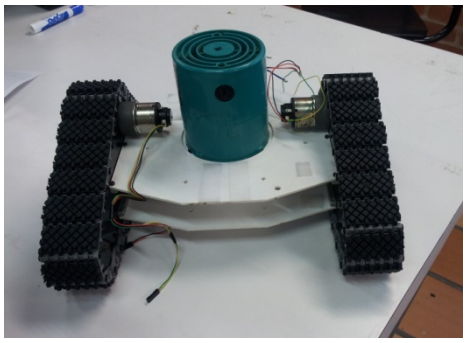

Fig. 22. Diseño en CAD de robot con oruga

6. DETERMINACIÓN DE LA TRAYECTORIA DEL ROBOT MÓVIL

El robot móvil debe explorar la superficie vertical, este recorrido puede ser ascendente y descendente desplazándose de forma horizontal al terminar cada barrido, o simplemente determinar algunos puntos de la superficie que fueron determinados como críticos o que se desean explorar para limpiar o analizar su estado. Por tal razón se implementó un algoritmo genérico que optimizara el recorrido del robot móvil para su navegación. En este ejemplo específico tomamos 17 puntos por el cual queremos que transite el robot las coordenadas están en orden totalmente aleatorio.

Posiciones(x,y)

X	0	1	4	9	12	10	6	9	18
Y	0	2	3	8	0	3	8	32	19

X	22	17	3	22	6	19	28	4	
Y	15	28	9	8	11	20	19	3	

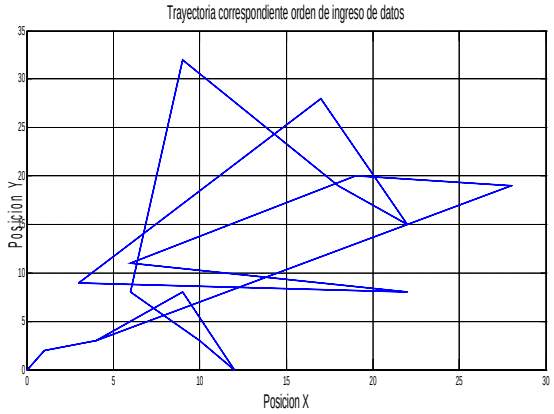

Fig. 23. Trayectoria correspondiente al orden de ingreso de las posiciones a recorrer.

El algoritmo genético evalúa de manera estocástica las posibles combinaciones las cuales son comparadas en función de la distancia total recorrida, de esta manera genera un numero de iteraciones que se ven limitadas por la solución del camino más corto o por el numero que se define por el programador en este caso se hace uso de 110 iteraciones que corresponden a un número superior al promedio para solución de prácticas que incorporen hasta 20 puntos en la trayectoria. Se define una población de 12 suficiente para evaluar las posibilidades de manera correcta.

Al poner en marcha el programa obtenemos los siguientes resultados:

En primera instancia con el parámetro *a=meshgrid(1:n)*; donde se genera una matriz de magnitudes 17 por 17 correspondiente al número de puntos de manera ordenada.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

Luego los vectores son ordenados de forma aleatoria con el fin de promover la mayor variación y crear la mayor cantidad de posibilidades teniendo en cuenta múltiples situaciones

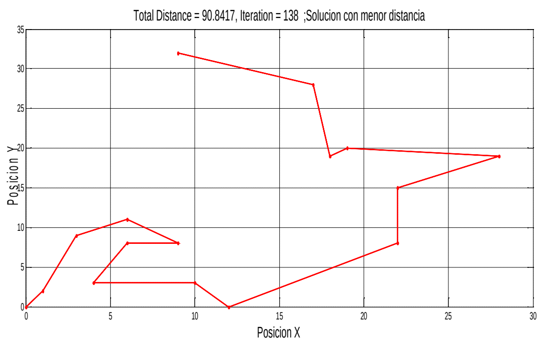

Fig. 24. Respuesta obtenida con el algoritmo genético.

Sin embargo es preciso especificar que en ocasiones el algoritmo no entrega la mejor respuesta y se debe correr varías veces el programa buscando reducir la distancia.

La Fig. 25 presenta la primera solución obtenida por el algoritmo se tiene una distancia de 96.3, con 196 iteraciones.

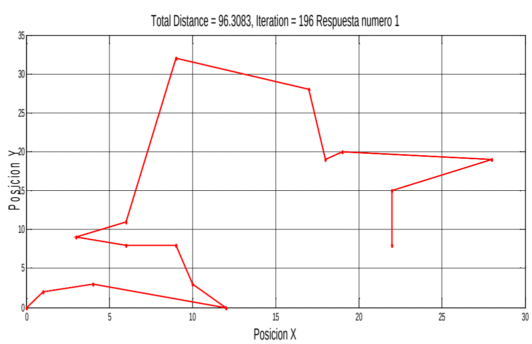

Fig. 25. Primera respuesta del algoritmo genético.

En la Fig. 26 al aumentar el número de iteraciones se mejora la primera respuesta encontrada en la primera prueba, obteniendo una distancia de 85.27 con 397 iteraciones.

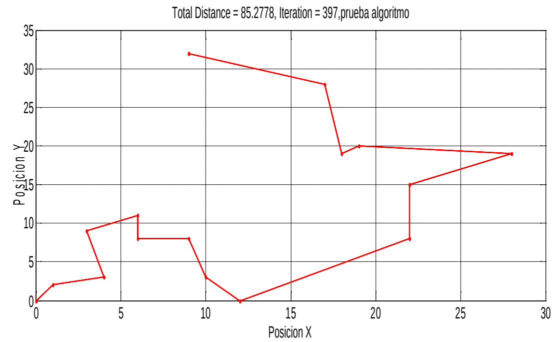

Fig. 26. Segunda respuesta del algoritmo genético.

Para mejorar la respuestase decide mantener el número de iteraciones en 500, para obtener una trayectoria más adecuada. Son generados por el software los ángulos necesarios para reproducir en la realidad dicho camino así como las distancias que deben mantener entre punto y punto y realizar la trayectoria.

Ángulos	0	63,4	18,43	180	99,4	33,6
Distancia	0	2,2	3,16	0	6,08	3,6

Ángulos	288	360	281,3	303,7	38,6	81,8
Distancia	3	3	5,09	3,6	12,8	7

Ángulos	33,7	173,7	225	96,3	153,4	
Distancia	7,21	9,05	1,41	9,05	8,94	

7. CONCLUSIONES

La implementación del encoder como sensor de velocidad, requiere de ventanas de tiempo para realizar el conteo de los pulsos. Esto hace que cada acción de control sea tomada cada 200 ms demorando un poco el tiempo de estabilización del sistema.

La implementación del algoritmo genético solo ofrece una idea para determinar cuál será el camino que debe seguir el robot móvil, sin embargo es necesario saber con qué orientación se debe realizar la inspección o limpieza en cada uno de los determinados.

Trabajar sobre superficies verticales incrementa las posibilidades de deslizamiento de cada una de las ruedas por tal razón se deben implementar sensores complementarios como brújula y acelerómetros para poder corregir estos problemas de ubicación relativa.

REFERENCIAS

Zhang H., Zhang J., Wei Wang, W., Liu R., Zong G., (2007), *A series of pneumatic glass-wall cleaning robots for high-rise buildings*, Industrial Robot: An International Journal, pp. 150–160.

Longo, D., Muscato, G., Sessa S., 2005, *Simulation and locomotion control for the Alicia3 climbing robot.* 22nd International Symposium on Automation and Robotics in Construction, ISARC 2005.

Nagakubo A., Hirose S., (1994), *Walking and running of the Quadruped Wall-Climbing Robot*, Proc. IEEE Int. Conf. on Robotics and Automation, pp. 1005-1012.

González H, Sarmiento, F., Lengerke, O., (2012). *Design of a Service Robot to Clean Storage Tanks*, Lecture Notes in Information Technology, Vol. 15, Materials, Mechatronics and Automation, pp 502-508.

Garrido. S., Moreno. L., (2003), *Ingeniería de control: modelado, análisis y control de sistemas*", Editorial Ariel S.A.