

Análisis comparativo de trayectorias en robots

1st Francisco Valdepeña
Tecnológico Nacional de México/CENIDET.
 Cuernavaca, Morelos, México
 francisco.valdepena19ca@cenidet.edu.mx

2nd Dante Mújica-Vargas
Tecnológico Nacional de México/CENIDET
 Cuernavaca, Morelos, México
 dantemv@cenidet.edu.mx

3rd Miguel Ángel Ruíz Jaimes
Universidad Politécnica del Estado de Morelos
 Jiutepec, Morelos
 mruiz@upemor.edu.mx

Resumen—En este artículo se analiza la planificación de trayectorias para robots, empleando los algoritmos: Control simple de trayectorias, Control por una trayectoria de referencia, Control basado en la optimización del enjambre de partículas y Control predictivo basado en modelos. Se realizó una evaluación cuantitativa basada en el Error Cuadrático Medio (ECM), Similitud coseno y el Coeficiente de Correlación de Pearson (CCP). Los sucesos experimentales indican que el Control predictivo basado en modelos, se desempeña mejor a trayectorias no lineales, con $ECM < 0.1$, Similitud coseno > 0.8 y $CCP < 0.02$, en un rango continuo de [0-1].

Index Terms—Análisis de trayectorias, Robots móviles, Algoritmos de control de trayectoria

I. INTRODUCTION

En la robótica móvil, una trayectoria es un vector de poses en un robot móvil necesita para manejar sobre coordenadas generalizadas en el espacio.

$$\mathbf{q}_{ref}(t) = [x_{ref}(t), y_{ref}(t), \varphi_{ref}(t)]^T \quad (1)$$

La mayoría de veces la trayectoria siempre se define en un intervalo de tiempo finito $t \in [0, T]$, es decir una trayectoria tiene un punto de inicio y un punto fin.

El control para el seguimiento de una trayectoria es un mecanismo que asegurará que la trayectoria del robot $\mathbf{q}(t)$ está lo más cerca posible de la referencia $\mathbf{q}_{ref}(t)$ a pesar de dificultades encontradas.

Al realizar un control de seguimiento de trayectoria, lo más lógico es imaginar la trayectoria de referencia como un movimiento a seguir.

En este documento se realiza un análisis comparativo del control de trayectorias sobre los algoritmos: Control basado en la optimización de ejambre de partículas (PSO), Control predictivo basado en modelos (MPC), Control simple de trayectorias y Control para una trayectoria de referencia [1]. En la sección 2, se describen conceptos sintetizados de la teoría del modelado de sistemas móviles, junto con su formalización matemática y los algoritmos de control de trayectorias a evaluar. En la sección 3, se desglosa la experimentación, métricas a emplear y los resultados del análisis. En la sección 4, se emplea una optimización al algoritmo con mejor desempeño. En la sección 5, se da una conclusión del análisis.

II. MARCO TEÓRICO

II-A. Modelado de robots móviles

La locomoción es el proceso de mover un sistema autónomo de un lugar a otro. Los modelos de movimiento pueden describir la cinemática del robot, esto es interesante ya que puede ser descrito matemáticamente, tal y como se muestra en la siguiente expresión [1].

$$\mathbf{q}(t) = \begin{bmatrix} x(t) \\ y(t) \\ \varphi(t) \end{bmatrix} \quad (2)$$

donde, $\mathbf{q}(t)$ es la trayectoria del robot, $x(t)$ y $y(t)$ son coordenadas locales en los ejes X e Y , $\varphi(t)$ es la orientación del robot.

En un modelado cinemático el vector pose de una trayectoria en un robot se describe geométricamente, por una matriz de rotación ($\mathcal{R}(\varphi)$), definida en la siguiente expresión [1].

$$\mathcal{R}(\varphi) = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

donde, cada columna de la matriz representa una relación entre las coordenadas locales del robot ($[x(t), y(t), z(t)]$) y las coordenadas globales de referencia ($[x_{ref}(t), y_{ref}(t), z_{ref}(t)]$), en un intervalo de tiempo.

La pose de un robot de configuración diferencial, sobre un plano de coordenadas globales está representada en la Figura 1.

II-B. Control de robots móviles

El control de la pose de un robot esta basada a una pose de referencia, la cual no esta predefinida y el robot puede elegir cualquier ruta factible para llegar a la meta. Sin embargo, esta ruta puede ser definida explícitamente y adaptada durante el movimiento o ser implícita aplicando un algoritmo de control de trayectorias, para lograr las poses de referencia [2].

En una pose, la orientación indica la disparidad a una referencia, esto en la mayoría de algoritmos es manejado mediante un error de estimación, entre la orientación de referencia y la

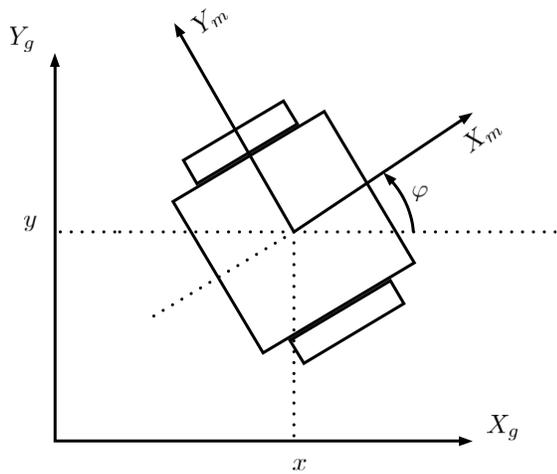


Figura 1. Descripción geométrica de la pose de un robot de configuración diferencial.

calculada por el algoritmo de control, tal y como se muestra en la siguiente expresión.

$$e_{\varphi}(t) = \varphi_{ref}(t) - \varphi(t) \quad (4)$$

donde, $\varphi_{ref}(t)$ orientación deseada, $\varphi(t)$ orientación del robot en relación, todo en función del tiempo.

Los tipos de algoritmos de control de trayectorias son:

- Control basado en la optimización de enjambre de partículas (PSO).
- Control predictivo basado en modelos (MPC).
- Control simple de trayectorias.

II-B1. Control basado en la optimización de enjambre de partículas.:

Se encuentra inspirado en el comportamiento de grupos (enjambre) de animales (partículas), donde cada animal tiene su propia hipótesis solución. Cada partícula i es caracterizada por un vector parámetro x_i que define su posición en un espacio y donde incrementa, el vector velocidad definido por v_i en cada espacio de parámetros ([3]).

Durante la optimización de la población todas las posibles soluciones (poses), son actualizadas, acorde a una función objetivo que define una medida de calidad [4].

Algorithm 1 Control basado en optimización enjambre de partículas

```

1: for  $i = 1 \rightarrow N$  do
2:    $x_i = random()$ 
3:    $v_i = 0$ 
4:    $pBest_i = x_i$ 
5:    $j_{best} = \infty$ 
6: for  $i = 1 \rightarrow N$  do
7:    $J_i = f(x_i)$ 
8:   if  $J_i < f(pBest_i)$  then
9:      $pBest_i = x_i$ 
10:  if  $f(pBest_i) < j_{best}$  then
11:     $gBest = pBest_i$ 
12:     $j_{best} = f(gBest)$ 
13: for  $i = 0 \rightarrow N$  do
14:   $v_i = \omega v_i + c_1 random(0,1)(pBest_i - x_i) + c_2 random(0,1)(gBest - x_i)$ 
15:   $x_i = x_i + v_i$ 
    
```

II-B2. Control predictivo basado en modelos (MPC):

En este modelo la trayectoria de referencia debe ser clara, pero suave para no generar ruido durante la ejecución. La predicción es un error linealizado y utiliza el modelo dinámico obtenido alrededor de la trayectoria de referencia. El controlador minimiza la diferencia entre el error de seguimiento futuro del robot y el error de referencia ([4]). Este control está basado en combinación de estrategias de solución previa o feedforward y una acción de retroalimentación (feedback) donde queda expresada de la siguiente manera [5].

$$u = u_{ff} + u_{fb} = \begin{bmatrix} v_{ref} \cos e_{\varphi} + v_{fb} \\ \omega_{ref} + \omega_{fb} \end{bmatrix} \quad (5)$$

donde, la entrada (feedforward) u_{ff} es calculada de la trayectoria de referencia y la u_{fb} (feedback) es la salida del controlador.

El problema de control viene dado por la dinámica de error de seguimiento lineal donde su notación queda expresada de la siguiente forma.

$$\hat{e} = A_c(t)e + B_c u_{fb} \quad (6)$$

donde, $A_c(t)$ y B_c son matrices del modelo continuo estado-espacio y e es el error de la trayectoria local del robot definida por la siguiente transformación.

$$e = \begin{bmatrix} \cos(\varphi(t)) & \sin(\varphi(t)) & 0 \\ -\sin(\varphi(t)) & \cos(\varphi(t)) & 0 \\ 0 & 0 & 1 \end{bmatrix} (q_{ref}(t) - q(t)) \quad (7)$$

donde, cada columna de la transformación indica la relación de orientación en los ejes $[x(t), y(t), z(t)]$, multiplicado por la diferencia entre la trayectoria generada y la de referencia.

II-B3. Control simple de trayectorias: Implementa un control básico, además del punto de referencia (x_{ref}, y_{ref}) requiere la orientación φ_{ref} donde el punto intermedio se calcula, mediante la siguiente expresión.

$$x_t = x_{ref} - r \cos \varphi_{ref}, \quad y_t = y_{ref} - r \sin \varphi_{ref} \quad (8)$$

donde, x_t es la coordenada local en X del robot e y_t en Y , todo en función del t (tiempo), diferencia de un radio r o distancia con respecto al ángulo de la pose del robot.

Al llegar al punto intermedio se evalúa la condición $(\sqrt{(x - x_t)^2 + (y - y_t)^2}) < d_{tol}$, siendo r la distancia entre punto medio y pose de referencia y d_{tol} la distancia para realizar el cambio de orientación. Para desarrollar la evaluación la trayectoria de referencia no-lineal tiene un dominio en X de 0.4 a 1.8 m (metros) con relación en Y de 0.2 a 1.6 m , tomando como posición inicial las coordenadas $[1, 1]$, tal y como se muestra en la Figura 2.

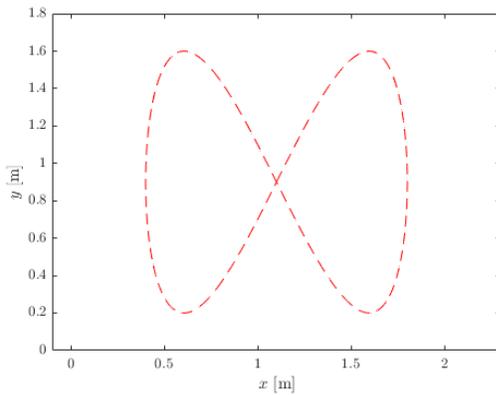


Figura 2. Trayectoria referencia no-lineal.

III. EXPERIMENTACIÓN Y RESULTADOS

El control en robots es un problema común a solucionar y varios de los algoritmos que se implementan en trayectorias lineales, no tienen buen desempeño en rutas no-lineales. Con el objetivo de generar una evaluación de los algoritmos de control mencionados, en esta experimentación se desarrolla una simulación para conocer el desempeño en trayectorias no-lineales.

El análisis comparativo, está basado en una evaluación del Error cuadrático medio (ECM), Similaridad en base al coseno del ángulo y Disimilaridad empleando el coeficiente de correlación de $r_{pearson}$.

Con el fin de evaluar, cuál algoritmo de control se comporta mejor con un ruido del 0.5% sobre trayectorias de tipo no-lineal.

El análisis comparativo se realizó sobre la plataforma MATLAB 2017b, bajos las siguientes características de hardware:

- Procesador: intel i3-7020U.
- RAM: 8 Gb.

- Arquitectura: 64 bits.

Al ejecutar cada algoritmo bajo una trayectoria de referencia $(q_{ref}(t))$, se obtuvieron diferentes controles de trayectoria, haciéndose más notorio con nivel de ruido del 0.05%, estos resultados, se observan en la Figura 3.

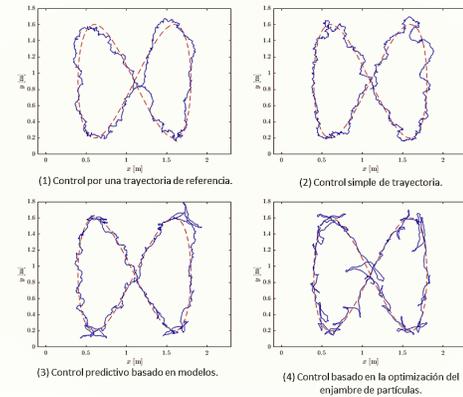


Figura 3. Algoritmos de control de trayectorias con un ruido de 0.5%.

III-A. Métricas de evaluación

III-A1. Similaridad entre vectores en base al ángulo del coseno: Es una medida de similitud entre dos vectores, con el que se evalúa el valor coseno entre ambas características [6].

$$Scos(x, y) = \frac{X^T Y}{\|X\| \|Y\|} \quad (9)$$

donde, $X^T Y$ es producto punto entre X e Y , $\|.\|$ es la magnitud del vector, resultando equivalente a la distancia euclidiana entre el origen y el vector X o Y .

III-A2. Error Cuadrático Medio: Es una función de riesgo, correspondiente al valor esperado de la pérdida del error al cuadrado o perdida cuadrática. Su diferencia es producida a la aleatoriedad o debido a que no toma en cuenta información para producir una estimación más precisa [7].

$$ECM = \frac{1}{n} \sum_{i=1}^n (\hat{X}_i - X_i)^2 \quad (10)$$

donde, \hat{Y}_i indica las n predicciones e Y los valores verdaderos.

III-A3. Coeficiente de Correlación r-Pearson: Es un índice utilizado para medir el grado de relación de dos variables siempre y cuando sean cuantitativas y continuas [8].

$$r_{pearson}(x, y) = \frac{\hat{X}^T \hat{Y}}{\|\hat{X}\| \|\hat{Y}\|} \quad (11)$$

donde, $Y = \{X_1, X_2, \dots, X_k\}$ e $Y = \{Y_1, Y_2, Y_3, \dots, Y_k\}$ son vectores, al igual que a y b , el valor de k es la longitud de los vectores correspondientes. Esta puede ser una medida de disimilaridad, como se muestra en la siguiente expresión.

$$CCP(x, y) = \frac{1 - r_{pearson}(x, y)}{2} \quad (12)$$

Durante el intervalo de tiempo de la simulación, el control de la trayectoria generado por el algoritmo, almacena un vector de poses $q(t)$ locales del robot con respecto a una referencia. Estos datos se clasifican en los componentes principales de una pose (x, y, φ) . De este modo, realizar un análisis es tomando una característica de la pose y evaluar dicho vector de datos con respecto a una referencia.

Cada vector de datos es evaluado en cada una de las métricas, con su componente correspondiente. El resumen del análisis

Tabla I
CONTROL SIMPLE DE TRAYECTORIAS.

	MSE	Coseno del ángulo	Coficiente de Pearson
x	0.009696	0.996954	0.007955
y	0.011657	0.994862	0.011609
φ	3.879206	0.401182	0.299170

Tabla II
CONTROL POR UNA TRAYECTORIA DE REFERENCIA

	MSE	Coseno del ángulo	Coficiente de Pearson
x	0.035708	0.989638	0.029632
y	0.063214	0.974061	0.030362
z	3.753112	0.463112	0.2526

Tabla III
CONTROL BASADO EN LA OPTIMIZACIÓN DEL ENJAMBRE DE PARTÍCULAS.

	MSE	Coseno del ángulo	Coficiente de Pearson
x	0.007116	0.997574	0.007418
y	0.008094	0.996242	0.008048
z	1.120763	0.819870	0.088982

Tabla IV
CONTROL PREDICTIVO BASADO EN MODELOS.

	MSE	Coseno del ángulo	Coficiente de Pearson
x	0.005208	0.998368	0.004811
y	0.026058	0.991004	0.016577
z	0.255301	0.959813	0.020056

de los algoritmos se representan en forma de tabla y se muestra en la parte de arriba. Para cada resultado cuantitativo hubo distintas muestra. Para el Control por trayectoria de referencia se manejaron 910 poses durante la simulación, en el Control basado en la optimización de enjambre de partículas 909 poses, el Control simple de trayectorias 1001 poses y en el Control predictivo basado en modelos 906 poses. Debido a la diferencia de poses en la trayectoria de referencia.

IV. FILTRADO DE TRAYECTORIA MEDIANTE K-NEAREST NEIGHBOR

Es un método de clasificación supervisado, que estima en base a un conjunto de entrenamiento, el valor a posteriori de que un elemento x pertenezca a una clase C_i .

Se dice que es un algoritmo de aprendizaje supervisado, debido que a partir de un conjunto de datos iniciales, se tiene como objetivo clasificar todas las instancias nuevas. Mediante

el cálculo de la distancia de un elemento nuevo a cada uno de los existentes, ordena dichas distancias de menor a mayor para ir seleccionando a la clase que pertenece. Por tanto, el de mayor frecuencia y con menor distancias, es el dato que clasifica con respecto al elemento.

El uso de este método como filtro de una señal, en este caso del control de la trayectoria en tiempo de ejecución, se esta explica en el siguiente algoritmo.

Algorithm 2 Filtro K-Nearest Neighbor

Input: $q(t), k$: pose, valor k, $(1 \leq k \leq n)$.

Output: $q_f(t)$: pose filtrada

```

1:  $L = 0$ 
2: for  $i = 1 \rightarrow length(t) - 4$  do
3:   if  $L \leq 3$  then
4:      $q_{ref_x}(k) = q(1)$ 
5:      $q_{ref_y}(k) = q(2)$ 
6:      $sum1+ = q_{ref_x}(k)$ 
7:      $sum2+ = q_{ref_y}(k)$ 
8:   if  $L == 3$  then
9:      $\mu_1 = sum1/3$ 
10:     $\mu_2 = sum2/3$ 
11:    $L = L + 1$ 
12:   if  $\sqrt{(q(1) - \mu_1)} < 0.001$  then
13:      $q(1) = q(1)$ 
14:   else
15:      $q(1) = \mu_1$ 
16:   if  $\sqrt{(q(2) - \mu_2)} < 0.001$  then
17:      $q(2) = q(2)$ 
18:   else
19:      $q(2) = \mu_2$ 
20:    $q_f(t) = q(t)$ 

```

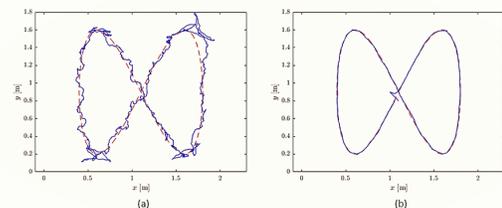


Figura 4. Control predictivo basado en modelos (a), control con el filtro K-Nearest Neighbor (b).

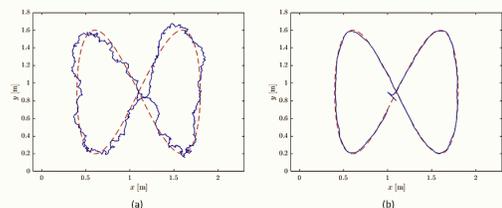


Figura 5. Control por una trayectoria de referencia (a), control con el filtro K-Nearest Neighbor (b).

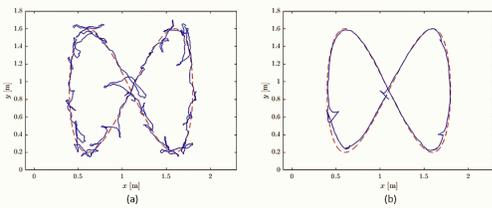


Figura 6. Control basado en la optimización de enjambre de partículas (a), control con el filtro K-Nearest Neighbor (b).

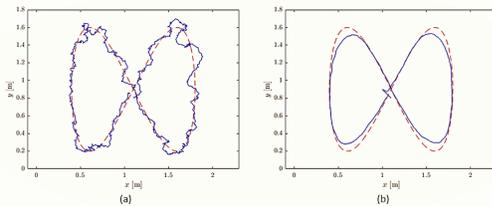


Figura 7. Control simple de trayectorias (a), control con el filtro K-Nearest Neighbor (b).

Los resultados del filtrado sobre cada algoritmo de control con un ruido de 0.5% se muestran, en la Figura 4-5-6-7. El nivel de ruido generado en la simulación solo refleja la posibilidad de que el robot no tenga un buen control, pero también cabe mencionar que al aplicar un filtro este no elimina el ruido o perturbación generada, que pueda tener el robot, lo cual se puede observar en la Figura 6, donde es posible apreciar irregularidades que aún se presentan al planificar la trayectoria.

A fin de cuentas esto sigue siendo un acercamiento favorable para generar un buen control de trayectorias.

V. CONCLUSIONES

La evaluación demuestra una correlación, similitud y disimilitud entre el vector pose generado por el algoritmo y el vector pose de referencia. El error cuadrático medio indica que tanto es el error en cada dato predecido al esperado, mostrando que en la mayoría de los algoritmos la dirección en cada instante de la trayectoria se ve afectada un ángulo considerable a causa de los factores de ruido, provocando una no-correlación en los ejes x e y .

Resultando más estable un control basado en modelos (MPC) ya que no se presentó una disparidad notable en el ángulo de rotación durante la trayectoria.

Los algoritmos de control resultaron muy sensibles al incremento de ruido durante la simulación, por lo tanto el emplear el filtrado K-Nearest Neighbor fue posible reducir más del 60% de ruido generado, de este modo ajustar la salida de los algoritmos de control a una trayectoria de referencia, recalco la afirmación que el Control predictivo basado en modelos se desempeña mejor a trayectorias no-lineales.

AGRADECIMIENTOS

Los autores expresan su agradecimiento a CONACYT, así como al Tecnológico Nacional de México / CENIDET por el

apoyo económico otorgado para la realización del proyecto 5628.19-P llamado "Sistema embebido para asistencia de conducción basado en Lógica Difusa Tipo-2".

REFERENCIAS

- [1] G. Klančar, A. Zdešar, S. Blažić, and I. Škrjanc, Eds., *Wheeled mobile robotics From fundamentals towards autonomous systems*. Butterworth-Heinemann, 2017.
- [2] M. I. C. Murgía, R. S. Rodríguez, and J. V. Pineda, *Percepción visual aplicada a la robótica*. México: Alfaomega, 2015.
- [3] R. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," *IEEE Xplore Digital library*, p. 5, 2000.
- [4] S. Vichik and F. Borrelli, "Solving linear and quadratic programs with an analog circuit," *ELSEVIER*, vol. 70, pp. 160–171, 2014.
- [5] D. N. Wilke, S. Kok, and A. A. Greonwold, "Comparison of linear and classical velocity update rules in particle swarm optimization: notes on scale and frame invariance," *Wiley online library*, vol. 70, pp. 985–1008, 2006.
- [6] A. Singhal, "Modern information retrieval: a brief overview," *BULLETIN OF THE IEEE COMPUTER SOCIETY TECHNICAL COMMITTEE ON DATA ENGINEERING*, vol. 24, p. 2001, 2001.
- [7] D. D. Wackerly, W. M. III, and R. L. Scheaffer, *Mathematical Statics with Applications 7th Edition*. USA: THOMSON, 2008.
- [8] P. Vinuesa, "Correlación: Teoría y practica," p. 26, 2016.