



Simulación del modelo matemático del seguimiento de trayectoria de un robot unicycle basado en métodos numéricos

Simulation of the mathematical model of the trajectory tracking of a unicycle robot based on numerical methods

Simulação do modelo matemático do rastreamento de trajetória de um robô monociclo baseado em métodos numéricos

Fabian Israel Heredia-Moreno ^I
fabian.heredia@esepoch.edu.ec
<https://orcid.org/0000-0003-3645-5165>

Michael Estefania Jativa-Brito ^{II}
estefania.jativa@esepoch.edu.ec
<https://orcid.org/0000-0002-6394-2586>

Pedro Javier Cornejo-Reyes ^{III}
pedro.cornejo@esepoch.edu.ec
<https://orcid.org/0000-0001-8256-9732>

Segundo Fabián Siza-Moposita ^{IV}
segundo.siza@esepoch.edu.ec
<https://orcid.org/0000-0001-8036-6974>

Correspondencia: fabian.heredia@esepoch.edu.ec

Ciencias Técnicas y Aplicadas
Artículo de Investigación

* **Recibido:** 22 de marzo de 2022 * **Aceptado:** 10 de abril de 2022 * **Publicado:** 12 de mayo de 2022

- I. Especialista de Proyectos, Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador.
- II. Docente de la Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador.
- III. Docente de la Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador.
- IV. Docente de la Escuela Superior Politécnica de Chimborazo, Riobamba, Ecuador.

Resumen

El presente artículo trata de la simulación del modelo matemático del seguimiento de trayectoria de un robot unicycle basado en métodos numéricos, dentro del cual se realiza un análisis explicativo y demostrativo del tema planteado.

El desarrollo se centra en el modelo cinemático con el punto de control desplazado y la aplicación de Métodos Numéricos (Euler), para lo cual se inicia con el diseño de tres diferentes controladores con el tema mencionado, se especifica sus ecuaciones entorno a la ley que rige a dicho tema con la idea de tener un análisis teórico claro y conciso. A continuación, se da paso a la programación en Matlab, en donde se diseña el controlador por Métodos Numéricos para el seguimiento de trayectoria.

Palabras Clave: Matlab; Seguimiento de Trayectoria; Diseño; Controlador; Modelo Cinemático.

Abstract

This article deals with the simulation of the mathematical model of the path tracking of a unicycle robot based on numerical methods, within which an explanatory and demonstrative analysis of the issue is carried out.

The development focuses on the kinematic model with the displaced control point and the application of Numerical Methods (Euler), for which it begins with the design of three different controllers with the topic, their equations are specified around the law that governs this topic with the idea of having a clear and concise theoretical analysis. Next, we proceed to programming in Matlab, where the controller is designed by Numerical Methods for trajectory tracking.

Keywords: Matlab; Path Tracking; Design; Controller; Kinematic Model.

Resumo

Este artigo trata da simulação do modelo matemático do rastreamento de trajetória de um robô monociclo baseado em métodos numéricos, dentro do qual é realizada uma análise explicativa e demonstrativa do tema proposto.

O desenvolvimento foca no modelo cinemático com o ponto de controle deslocado e na aplicação de Métodos Numéricos (Euler), para o qual se inicia com o projeto de três controladores diferentes com o tópico supracitado, suas equações são especificadas em torno da lei que rege este tópico com a ideia de ter uma análise teórica clara e concisa. Em seguida, é apresentada a

programação em Matlab, onde o controlador é projetado por Métodos Numéricos para rastreamento de trajetória.

Palavras-chave: Matlab; Rastreamento de trajetória; Projeto; Controlador; Modelo cinemático.

Introducción

El robot tipo unicycle perteneciente a la categoría de robots móviles es un vehículo capaz de seguir una trayectoria que ha sido previamente planificada dentro de un espacio de trabajo. Estos autómatas son utilizados actualmente en diferentes áreas que requieran la intervención de este tipo de móviles, se debe tomar en cuenta que no todos los robots unicycle son configurados para seguir una trayectoria.

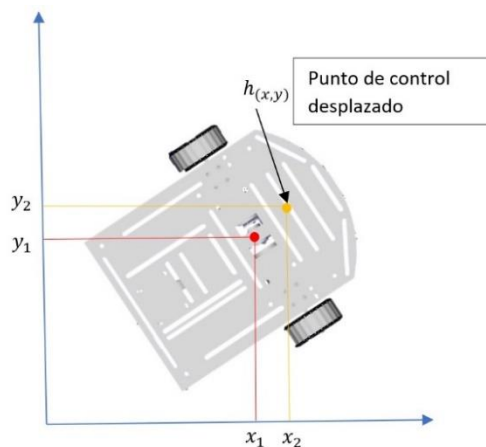
El presente artículo por habla del modelado y de la simulación de un robot unicycle, el objetivo principal es diseñar un controlador óptimo para un correcto seguimiento de trayectoria, esto en base a un análisis específico de cada paso a seguir.

Al final del trabajo se pretende obtener una respuesta clara entorno al seguimiento de trayectoria del robot unicycle, tomando a consideración el estudio y análisis del controlador diseñado.

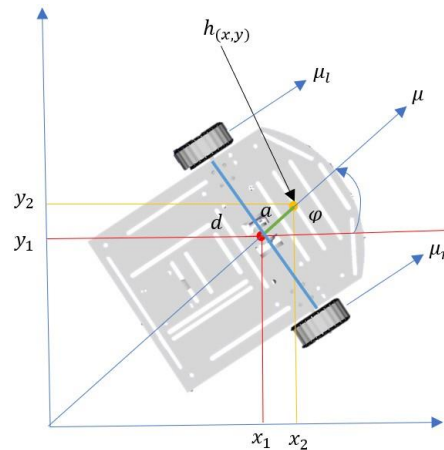
Metodología

Modelo cinemático con punto de control desplazado

El cálculo del modelo cinemático con el punto de control desplazado nos permite implementar manipuladores para futuras investigaciones.



De esta manera el objetivo es hallar el modelo con respecto al punto de control desplazado. Determinamos μ_l velocidad lineal de la rueda derecha, μ_r velocidad lineal de la rueda izquierda el ángulo de φ orientación y la posición del robot $h(x, y)$.

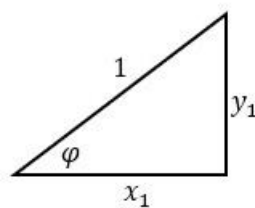


Observamos en la figura que geoméricamente en el eje x y y se obtiene

$$h_x = x_1 + x_2$$

$$h_y = y_1 + y_2$$

Para hallar los puntos en la posición x_1 y y_1



$$\cos(\varphi) = \frac{x_1}{1}, \sin(\varphi) = \frac{y_1}{1}$$

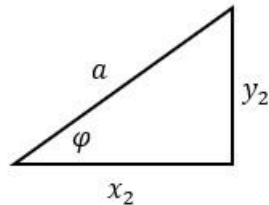
$$x_1 = \cos(\varphi), y_2 = \sin(\varphi)$$

Se deriva la posición y se obtiene la velocidad μ , Por consiguiente, la velocidad se va a descomponer en el eje (x, y) por lo tanto se tendrá que la velocidad en el punto x será igual a la velocidad lineal.

$$\dot{x}_1 = \mu \cos(\varphi)$$

$$\dot{y}_1 = \mu \sin(\varphi)$$

Para hallar las ecuaciones del punto deseado, observamos que se forma un triángulo rectángulo en la cual aplicamos razones trigonométricas y se obtiene:



$$\cos \varphi = \frac{x_2}{a}, \sin \varphi = \frac{y_2}{a}$$

Despejando x_2 y y_2

$$x_2 = a \cos(\varphi), y_2 = a \sin(\varphi)$$

Derivamos el sistema de ecuaciones con respecto al tiempo utilizando la regla de la cadena, y obtenemos que la derivada de la orientación $\dot{\varphi}$ es la velocidad angular ω .

$$\dot{x}_2 = -a\omega \sin(\varphi)$$

$$\dot{y}_2 = a\omega \cos(\varphi)$$

Remplazamos en el sistema de ecuaciones

$$\dot{\varphi} = \omega$$

$$\dot{h}_x = \mu \cos(\varphi) - a\omega \sin(\varphi)$$

$$\dot{h}_y = \mu \sin(\varphi) + a\omega \cos(\varphi)$$

Ahora se procede a dejar el sistema de ecuaciones en función de las velocidades lineales de la rueda izquierda y derecha, para esto encontramos la velocidad total del sistema que será el promedio de la velocidad de la llanta derecha μ_r más promedio de la velocidad de la llanta izquierda.

$$\mu = \frac{\mu_r + \mu_l}{2}, y$$

La velocidad angular que se relaciona con la velocidad lineal de la llanta derecha menos la velocidad angular de la llanta izquierda por una distancia d .

$$\omega = \frac{\mu_r - \mu_l}{d}$$

Remplazamos en el sistema de ecuaciones

$$\dot{h}_x = \frac{\mu_r + \mu_l}{2} \cos(\varphi) - a \frac{\mu_r - \mu_l}{d} \sin(\varphi)$$

$$\dot{h}_y = \frac{\mu_r + \mu_l}{2} \sin(\varphi) + a \frac{\mu_r - \mu_l}{d} \cos(\varphi)$$

$$\dot{\varphi} = \frac{\mu_r - \mu_l}{d}$$

Se procede a un planteamiento matricial del sistema de ecuaciones encontrado.

$$\begin{bmatrix} \dot{h}_x \\ \dot{h}_y \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \frac{\cos(\varphi)}{2} - \frac{a \sin(\varphi)}{d} & \frac{\cos(\varphi)}{2} + \frac{a \sin(\varphi)}{d} \\ \frac{\sin(\varphi)}{2} + \frac{a \cos(\varphi)}{d} & \frac{\sin(\varphi)}{2} - \frac{a \cos(\varphi)}{d} \\ \frac{1}{d} & -\frac{1}{d} \end{bmatrix} \begin{bmatrix} \mu_r \\ \mu_l \end{bmatrix}$$

Una vez que ha obtenido la matriz Jacobiana, finalmente se tiene el modelo matemático el cual relaciona las velocidades en el punto de control con las velocidades de los actuadores.

Diseño del Controlador basado en Métodos Numéricos

Si consideramos la ecuación diferencial de primer orden

$$\dot{y} = f(y, u, t) \quad y(0) = y_0$$

El uso de métodos numéricos en el cálculo de la evolución de sistemas se basa en la posibilidad de aproximar el estado del sistema en el instante de tiempo $k + 1$, si el estado y la acción de control en el instante de tiempo k son conocidas, esta aproximación se denomina método de Euler.

$$y(k + 1) = y(k) + t_s f(y, u, t)$$

De esta forma el modelo discreto puede ser expresado

$$y(k + 1) = y(k) + t_s J(q(k))\dot{q}(k)$$

$J(q(x))$ Matriz jacobiana

$\dot{q}(k)$ Vector de velocidades

El objetivo del control es llegar a una trayectoria deseada encontraremos el estado siguiente por el estado siguiente deseado agregando el error.

$$h(k+1) = h_d(k) - w(h_d(k) - h(k))$$

Donde w es una matriz diagonal con $0 < (w_x, w_y, w_z) < 1$ y h_d es la trayectoria deseada

Por lo tanto, se obtiene el sistema de ecuaciones de la forma

$$J(q(k))\dot{q}(k) = h_d(k) - w(h_d(k) - h(k)) - h(k)$$

Si consideramos que el sistema tiene el mismo número de ecuaciones e incógnitas $m = n$ las velocidades de los actuadores pueden obtenerse mediante la simple inversión de la matriz jacobina.

$$\dot{q}_{ref} = J^{-1}b$$

De esta manera encontraríamos la ley de control propuesta.

Ahora procedemos a realizar la estabilidad, partiendo de la aproximación del modelo en tiempo discreto, la ley de control

$$h(k+1) = h(k) + t_s J(q(k))\dot{q}(k)$$

$$\dot{q}_{ref} = J^{-1}b$$

Para hallar la ecuación en lazo cerrado consideramos el seguimiento perfecto de velocidades del controlador el cual va a ser iguales a las velocidades de lectura del robot unicycle que se obtiene mediante encoder.

$$\dot{q}_{ref} = \dot{q}(k)$$

$$h(k+1) = h(k) + t_s J J^{-1}b$$

Donde $J J^{-1}$ es matriz identidad

$$h(k+1) - h(k) = t_s b$$

$$h(k+1) - h(k) = t_s \left(\frac{h_d(k+1) - w(h_d(k) - h(k)) - h(k)}{t_s} \right)$$

$$h_{xd}(k+1) - h_x(k+1) = w_x(h_{xd}(k) - h_x(k))$$

$$h_{ex}(k+1) = w_x h_{ex}(k)$$

$$h_{ex}(k+2) = w_x^2 h_{ex}(k)$$

$$h_{ex}(k+3) = w_x^3 h_{ex}(k)$$

$$h_{ex}(k+n) = w_x^n h_{ex}(k)$$

Implementación del Controlador en Matlab

Encabezado. - en este punto se declara y se inicializa las variables como el tiempo de simulación (tf), tiempo de muestreo (ts), vector de tiempo(t), distancia al punto de interés en metros (a) y numero de muestras Q-1 esto es debido a que el algoritmo de control necesitamos el valor $hd(k+1)$.

```
clear
```

```
close all
```

```
clc
```

```
tf=40;
```

```
ts=0.1;
```

```
t=0:ts:tf;
```

```
a=0.1;
```

```
Q=length(t)-1;
```

Condiciones iniciales. – comprende la posición central inicial (eje x) en metros (m), punto de inicio del robot, posición real inicial (eje x) en metros (m), ángulo de orientación de partida (rad). Se debe tomar en cuenta que los parámetros el punto de inicio del robot con el punto de inicio de la trayectoria deseada deben ser iguales.

```
x1 = zeros(1,Q+1);
```

```
y1 = zeros(1,Q+1);
```

```
x1(1) = -0.07;
```

```
y1(1) = 0.5;
```

```
hx = zeros(1,Q+1);
```

```
hy = zeros(1,Q+1);
```

```
phi = zeros(1,Q+1);
```

```
phi(1)=-pi/2;
```

Modelo geométrico con el punto desplazado. - se establece la posición de interés inicial (eje x) en metros (m) y la posición de interés inicial (eje y) en metros (m), ingresamos el modelo cinemático con control de posición desplazado calculado en los temas anteriores.

```
hx(1) = x1(1)+a*cos(phi(1));
```

```
hy(1) = y1(1)+a*sin(phi(1));
```

Configuración de la Trayectoria deseada. - comprende el ingreso de la trayectoria que va a realizar el robot unicycle, trayectoria que depende del tiempo.

```
hxd=0.1*t;
```

```
hyd=0.5*cos(0.3*t);
```

Comportamiento del controlador. - Constantes que permiten regular el controlador, parámetros encontrados en el tema de diseño del controlador basado en métodos numéricos.

$w_x=0.90;$

$w_y=0.90;$

Enserando los vectores. - medición de la velocidad lineal (m / s), medición de la velocidad angular

$u_{Ref}=\text{zeros}(1,Q);$

$w_{Ref}=\text{zeros}(1,Q);$

$h_{xe} = \text{zeros}(1,Q);$

$h_{ye} = \text{zeros}(1,Q);$

$\phi_{ie} = \text{zeros}(1,Q);$

$\text{tiempo}=\text{zeros}(1,Q)$

Controlador. – posición de los errores el cual cambia en cada instante de muestreo de esta manera encontramos el error en los ejes x y y al final armaremos un vector de errores.

a) error

$h_{xe}(k) = h_{xd}(k) - h_x(k);$

$h_{ye}(k) = h_{yd}(k) - h_y(k);$

$h_e = [h_{xe}(k) \ h_{ye}(k)]';$

Luego se procede a ingresar la matriz jacobiana encontrada en la ecuación

b) Matriz jacobiana

$J_{11} = \cos(\phi(k));$

$J_{12} = -a*\sin(\phi(k));$

$J_{21} = \sin(\phi(k));$

$J_{22} = a*\cos(\phi(k));$

$J=[J_{11} \ J_{12};...$

J21 J22];

c) ingreso del parámetro b encontrada en el diseño del controlador en la ecuación.

$$b = (1/ts)*[hxd(k+1)-wx*hxe(k)-hx(k);... \\ hyd(k+1)-wy*hye(k)-hy(k)];$$

Como último punto se aplica ley de control encontrada en la cual las velocidades de referencia son iguales a la inversa de la matriz de la matriz jacobiana por la derivada de la trayectoria más la constante k por el vector de errores y procederemos a separar las velocidades lineal y angular.

d) Control de la ley

$$qpRef = pinv(J)*(b);$$

e) Control de acción independiente

$$uRef(k) = qpRef(1);$$

$$wRef(k) = qpRef(2);$$

Simulación del robot. - simulación del control de trayectoria del robot unicycle

% a) Robot simulado (modelo cinemático)

phi(k+1)=phi(k)+ts*wRef(k); % orientación real en radianes

x1p=uRef(k)*cos(phi(k+1)); % velocidad real en metros / segundos

y1p=uRef(k)*sin(phi(k+1)); % velocidad real en metros / segundos

% b) integral numérica (método de Euler)

x1(k+1)=x1(k)+ts*x1p; % posición real del centro (eje x) en metros

y1(k+1)=y1(k)+ts*y1p; % posición real del centro (eje y) en metros

% c) Modelo geométrico

x(k+1)=x1(k+1)+a*cos(phi(k+1));% posición de interés real (eje x) en metros (m)

y(k+1)=y1(k+1)+a*sin(phi(k+1));% posición de interés real (eje y) en metros (m)

Movimiento estroboscópico del robot unicycle. - en este apartado se llama al archivo robotParameters y robotPlot que son los parámetro del robot para realizar la simulación

```
scene=figure;
tam=get(0,'ScreenSize');
set(scene,'position',[tam(1) tam(2) tam(3) tam(4)]); % posición y
figura de tamaño en la pantalla
axis equal; % Establecer relaciones de aspecto del eje
axis([-5 5 -5 5 -0.5 1]); % Establecer límites de eje
view([135 35]); % figura de orientación
grid on;

robotParameters; % llamar al archivo parametros del robot
M1=robotPlot(x(1),y(1),phi(1)); %Trace el robot en la posición inicial
x1, y1 y orientación phi
hold on
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); %
camlight right % luz de la escena

M2=plot3(x(1),y(1),0.26,'go','LineWidth',2); % punto inicial
M3=plot3(x(1),y(1),zeros(1),'b','LineWidth',2); % trayectoria inicial
M4=plot3(hxd,hyd,zeros(1,Q),'r','LineWidth',2); % trayectoria inicial

step=10;

for i=1:step:Q % Loop emulationk
    delete (M1) % borrar trayectoria
    delete (M3) % borrar robot
    M1=robotPlot(x(i),y(i),phi(i)); hold on; % Trama de robot de nuevo
```

```
M3=plot3(x(1:i),y(1:i),zeros(1,i),'b','LineWidth',2); %
trayectoria de la trama.
pause(ts)
end
```

Resultados, Discusión y Análisis

Una vez concluida la programación en Matlab se procedió a modificar los valores para obtener los resultados del modelo matemático.

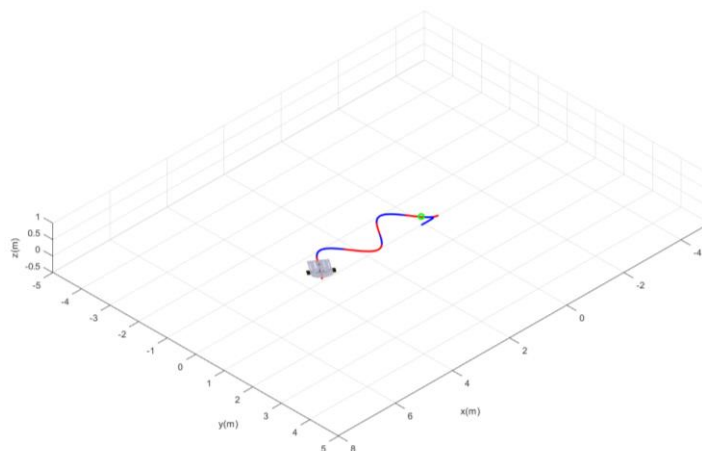
Validación Del Modelo

Una vez concluida la programación del control de seguimiento de trayectoria basa en métodos numéricos se procede a la verificación de la simulación

La trayectoria que se procede a verificar es una sinusoidal, los valores de entrada se inicializan con:

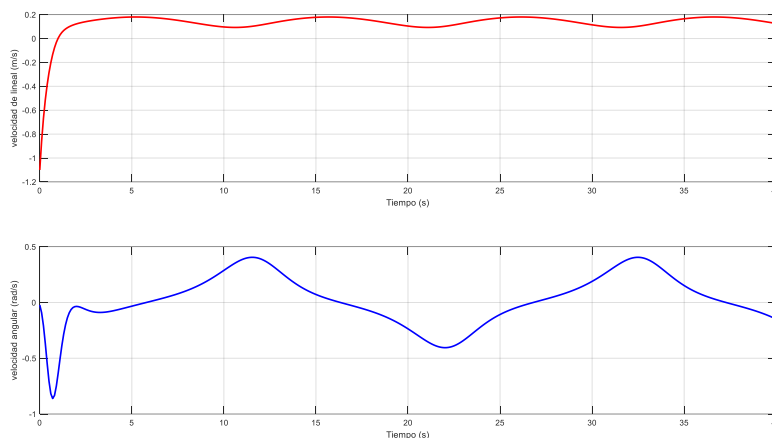
| Punto de partida del robot | Trayectoria | Parámetros de control | Tiempo de simulación | Angulo de orientación |
|----------------------------|-----------------------------------|-----------------------|----------------------|-----------------------|
| x1(1)=0.5, y1(1)=0.5 | hxd=0.1*t; hyd=0.5*cos(0.3*t); | wx=0.90; wy=0.90; | 40s | -pi/2 |

Trayectoria simulada



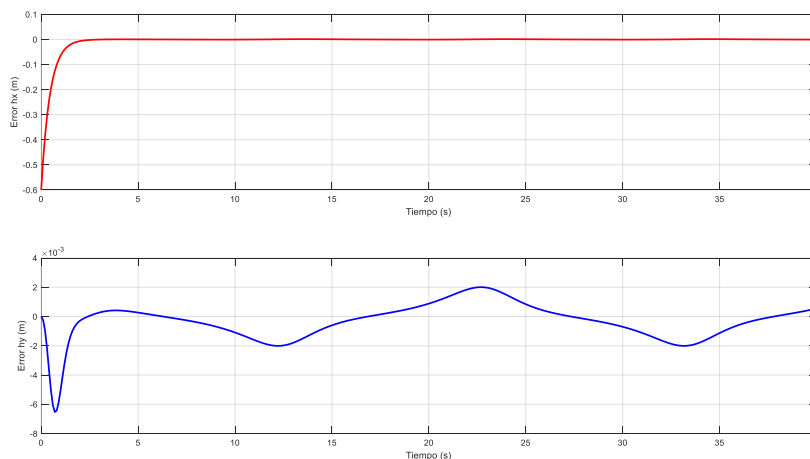
Simulación de trayectoria sinusoidal

A continuación, se representan las velocidades lineales y angulares al simular la trayectoria mencionada con los parámetros de control establecidos.



Velocidades lineales y angulares de trayectoria sinusoidal

En la figura se puede observar los errores al seguir la trayectoria

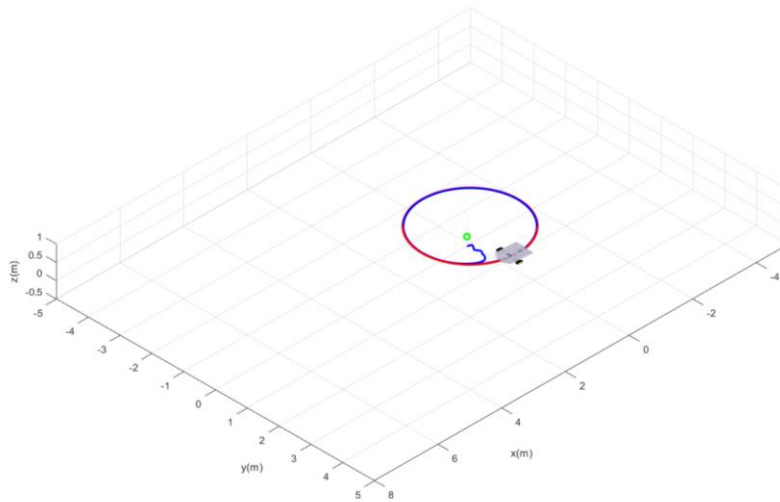


Verificación de errores en trayectoria sinusoidal

La segunda trayectoria a verificar su comportamiento es una trayectoria circular con los siguientes parámetros

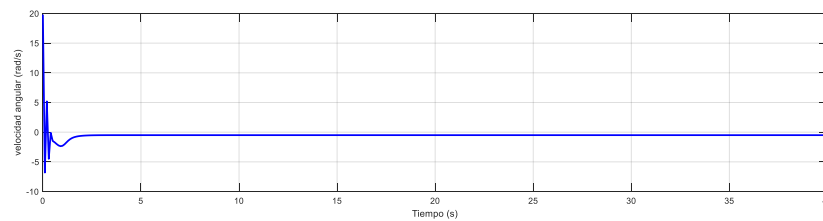
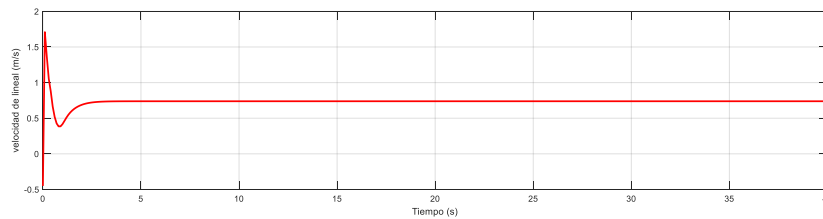
| Punto de partida del robot | Trayectoria | Parámetros de control | Tiempo de simulación | Angulo de orientación |
|-----------------------------|--|--------------------------|----------------------|-----------------------|
| $x1(1)=0.5,$ $y1(1)=0.5$ | $hxd=1.5*\sin(0.5*t)$ $hyd=1.5*\cos(0.5*t)$ | $wx=0.90;$ $wy=0.90;$ | 40s | $-\pi/2$ |

La trayectoria para simular se visualiza en la figura



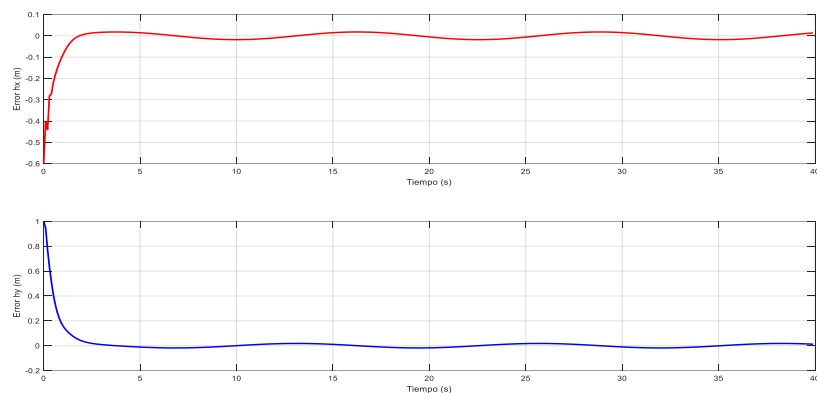
Simulación de trayectoria circular

Se presenta el comportamiento de las velocidades lineales y angulares



Velocidad angular y lineal trayectoria circular

En la figura se representa la evaluación de los errores al seguir la trayectoria

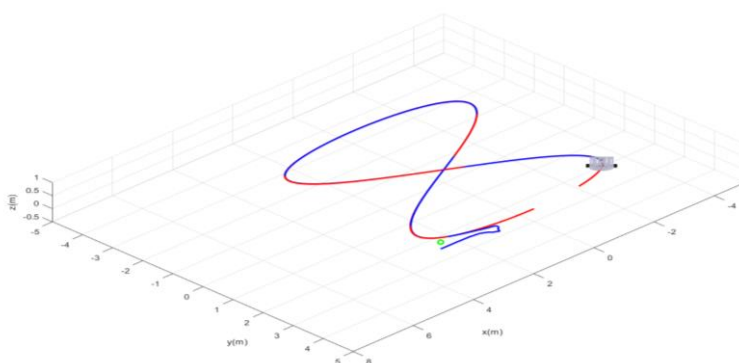


Verificación de errores trayectoria circulara

Y la última trayectoria a verificar es una trayectoria tipo lazo con los parámetros de entrada como se determina en la tabla:

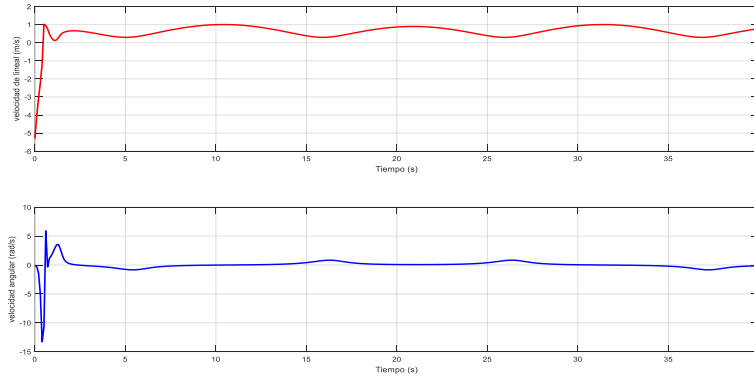
| Punto de partida del robot | Trayectoria | Parámetros de control | Tiempo de simulación | Angulo de orientación |
|----------------------------|---|--------------------------|----------------------|-----------------------|
| $x1(1)=3$ $y1(1)=3$ | $hxd=3*\sin(0.3*t)$ $hyd=3*\cos(0.15*t)$ | $wx=0.90;$ $wy=0.90;$ | 40s | $-\pi/2$ |

El seguimiento de la trayectoria tipo lazo se puede observar en la figura



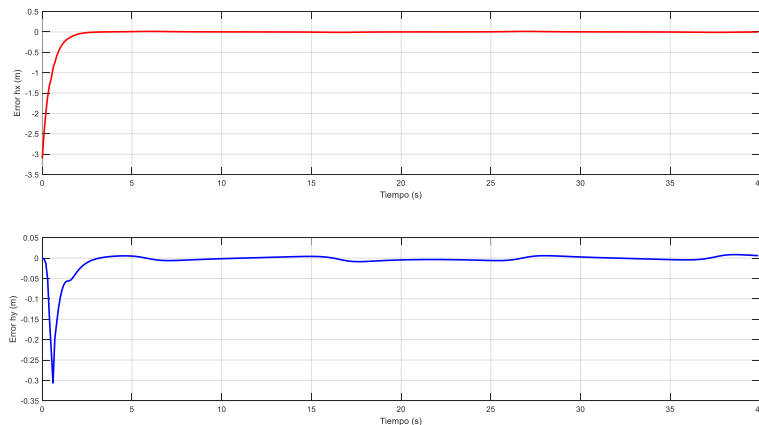
Simulación de trayectoria circular

En la figura se evidencia el comportamiento de las velocidades lineales y angulares



Velocidad angular y lineal trayectoria tipo lazo

En la figura se presentan los errores al seguir la trayectoria tipo lazo



Verificación de errores trayectoria tipo lazo

Análisis de datos

Una vez realizada la verificación de la implementación del controlador basado en métodos numéricos en el software Matlab se procedió a la variación de cada uno de los parámetros con el objetivo una respuesta y un registro de errores y velocidades.

Resultado de la simulación del control de seguimiento de trayectoria sinusoidal:

| Parámetro de regulación | Angulo de orientación Phi | Error inicial | | Valores máximos | |
|-------------------------|---------------------------|---------------|--------|------------------------|-----------------------------|
| | | Hx (m) | hy (m) | Velocidad lineal (m/s) | Velocidad angular (rad/seg) |
| wx=1.5; wy=1.5; | $-\pi/2$ | 5.40 | 2.98 | 23.42 | 139.7 |
| wx=1; wy=1; | $-\pi/2$ | 0.7 | 0.1 | 0.18 | 0.1 |
| wx=0.8; wy=0.8; | $-\pi/2$ | 0.07 | 0.1 | 0.12 | 2.5 |
| wx=-0.6; wy=-0.6; | $-\pi/2$ | 0.14 | 0.1 | 2.39 | 12.55 |

Para el análisis del controlador por Métodos Numéricos el parámetro de regulación debe cumplir con $0 > wx, wy < 1$, en la primera y la última simulación se ha tomado un valor excedente al intervalo, es decir mayor que 1 y menor que 0 respectivamente, con la finalidad de demostrar el comportamiento del robot como se observa los resultados en la tabla. Los parámetros de regulación de las dos siguientes simulaciones se encuentran dentro del intervalo $0 > wx, wy < 1$, en donde se puede determinar un comportamiento diferente, se identifica que el error de seguimiento de trayectoria disminuye además las velocidades lineales como angulares se pueden manejar.

Los errores al seguir la trayectoria son mínimos cuando el parámetro w se encuentra dentro del intervalo $0 > wx, wy < 1$ y se puede determinar que las velocidades lineal y angular son periódicas para la trayectoria indicada.

Resultado de la simulación del control de seguimiento de trayectoria circular:

| Parámetro de regulación | Angulo de orientación Phi | Error inicial máximo | | Valores máximos | |
|-------------------------|---------------------------|----------------------|--------|------------------------|-----------------------------|
| | | Hx (m) | hy (m) | Velocidad lineal (m/s) | Velocidad angular (rad/seg) |
| wx=1; wy=1; | $-\pi/2$ | -0.032 | 0.11 | 0.1 | -0.1 |
| wx=0.6; wy=0.6; | $-\pi/2$ | -0.05 | 0.11 | 0.1 | -0.1 |
| wx=-0.5; wy=-0.5; | $-\pi/2$ | 0.065 | 0.11 | 1.43 | 8 |

Se ha seleccionado el parámetro de regulación igual a uno para la primera simulación, con la idea de analizar en la tabla el comportamiento del robot al tomar el límite del intervalo de w , en donde los errores resultaron ser significativamente grandes al seguir la trayectoria.

En la tercera simulación se ha considerado que el parámetro de regulación se encuentra fuera del intervalo $0 > wx, wy < 1$, en donde se puede observar que existen cambios bruscos en sus velocidades lineal y angular, así como también al inicio de la trayectoria.

En la tabla se confirma que los parámetros wx, wy se deben encontrar entre el intervalo $0 > wx, wy < 1$ con el objetivo de disminuir el error de seguimiento de trayectoria y que las velocidades lineal y angular no presenten variaciones significativas.

Resultado de la simulación del control de seguimiento de trayectoria tipo lazo:

| Parámetro de regulación | Angulo de orientación Phi | Error inicial máximo | | Valores máximos | |
|-------------------------|---------------------------|----------------------|--------|------------------------|-----------------------------|
| | | Hx (m) | Hy (m) | Velocidad lineal (m/s) | Velocidad angular (rad/seg) |
| wx=1.2; wy=1.2; | $-\pi/2$ | 5.48 | 3.8 | 8.24 | 60 |
| wx=0.5; wy=0.5; | $-\pi/2$ | 0.094 | 0.0033 | 0.46 | 2.51 |
| wx=-0.1; wy=-0.1; | $-\pi/2$ | 17523 | 7422 | 14 | 25 |

En la trayectoria tipo lazo, al realizar la primera y última simulación con el parámetro w fuera del intervalo se determina que los errores y las velocidades lineal ya angular al seguir la trayectoria presentan variaciones considerables.

En la segunda simulación se determina que los puntos de partida del robot y de la trayectoria, así como los ángulos de dirección son iguales, además el parámetro de regulación se encuentra dentro del intervalo $0 > wx, wy < 1$, dando como resultado un comportamiento en torno a sus errores y velocidades idóneas al seguir la trayectoria.

Conclusiones

- Se ha de terminado que el punto de partida del robot unicycle tiene que ser igual al punto de partida de la trayectoria y de la misma manera el ángulo de trayectoria igual al ángulo de partida del robot.
- Se permitió identificar los valores de wx y wy con los cuales el robot unicycle puede completar la trayectoria con el menor número de errores, disminuyendo de esta manera las oscilaciones que se presentan al momento de la simulación.
- El control de seguimiento de trayectoria se ve afectado por la forma de la trayectoria seguir; además la mejor forma de garantizar el funcionamiento correcto de estos sistemas de control es mediante una trayectoria que no presente cambios abruptos de dirección.

Referencias

1. Cartagena, M., & Pillajo, R. (2018). DISEÑO, SIMULACIÓN Y COMPARACIÓN DE CONTROLADORES BASADOS EN MÉTODOS NUMÉRICOS Y ÁLGEBRA LINEAL APLICADOS A PROCESOS QUÍMICOS CON RESPUESTA INVERSA APROXIMABLES A MODELOS DE ORDEN REDUCIDO. Escuela Superior Politécnica Nacional.
2. Andaluz Ortiz, G. M. (2015). Escuela Superior Politécnica De Chimborazo.
3. Heredia, F., & Játiva, M. (2021). Modelado y simulación de un robot unicycle para seguimiento de trayectoria. Universidad Internacional de la Rioja.