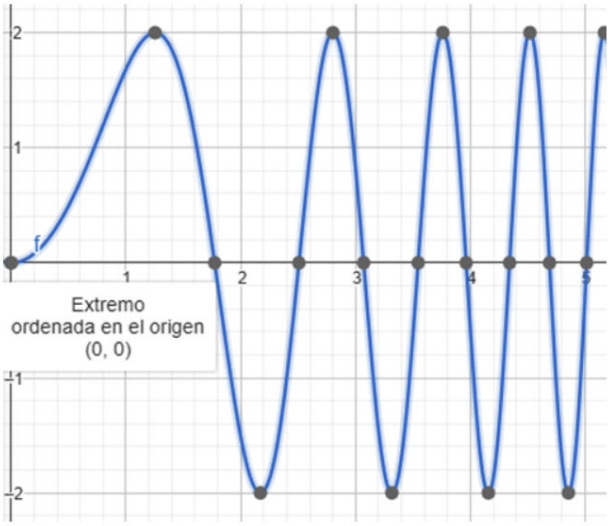
MARIAM LANDA BAUTISTA TRAYECTORIA 1

Describe la simulación de tres trayectorias diferentes realizadas con un robot móvil. Cada simulación utiliza un modelo gráfico 3D del robot y muestra su desplazamiento a lo largo de trayectorias personalizadas definidas matemáticamente. Las trayectorias permiten observar el comportamiento del robot en movimiento, así como su orientación (ángulo φ) en todo momento.

Trayectoria 1:

X= [0 a 5]

F(x) = 2*sen(x^2)

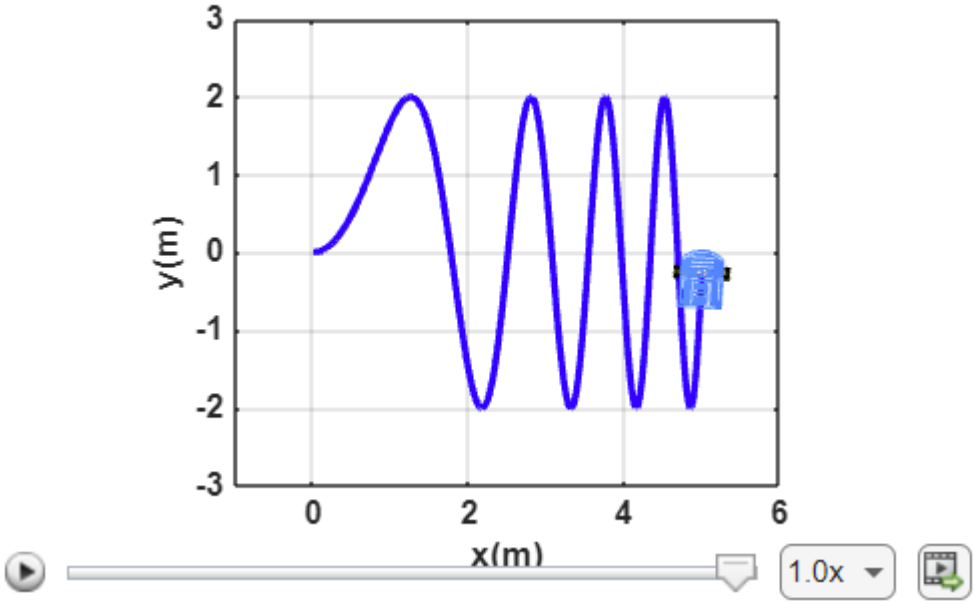


El primer script genera una trayectoria sobre el eje X en el intervalo [0,5][0, 5][0,5] y calcula los valores de Y utilizando la función y=2·sin(x2)y = 2 \cdot \sin(x^2)y=2·sin(x2). Esta trayectoria tiene un comportamiento oscilatorio que se intensifica conforme aumenta x. Es Una trayectoria ondulada, con variaciones rápidas a lo largo del eje Y conforme el robot avanza en X.

- Se genera el vector `x_range` con 300 puntos.
- Se calcula `phi_square` usando la derivada aproximada del ángulo de orientación.
- Se visualiza la trayectoria en una escena 3D con el robot animado.

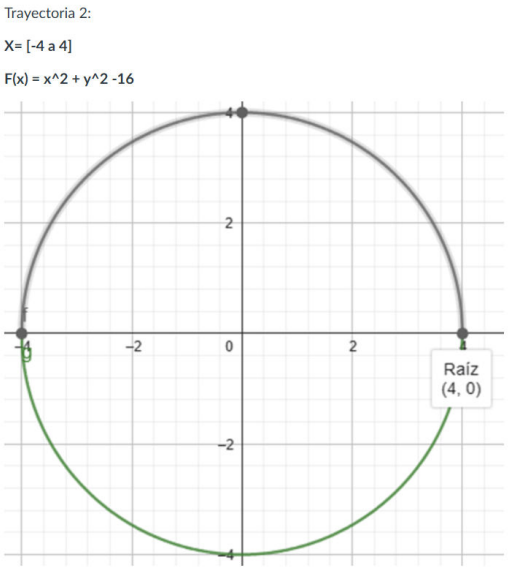
```
clear
close all
clc

%%%%%%%%%%%%%%%%%%%%%%%%%% TRAYECTORIA PERSONALIZADA %%%%%%%%%%%%%%%%%%%%%%%%%%

x_range = linspace(0, 5, 300);       % Vector de X
y_range = 2 * sin(x_range.^2);       % Y = 2·sin(x²)
```

```
hx_square = x_range;
hy_square = y_range;

%%%%%%%%%%%%%%%%%%%%% ORIENTACIÓN DEL ROBOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

phi_square = atan2(diff(hy_square), diff(hx_square));
phi_square(end+1) = phi_square(end);  % Igualar longitudes

x1_square = hx_square;
y1_square = hy_square;

N_square = length(x_range);
ts = 0.03;  % Tiempo de muestreo

%%%%%%%%%%%%%%%%%%%%%% SIMULACIÓN GRÁFICA %%%%%%%%%%%%%%%%%%%%%%
scene1 = figure;
set(scene1,'Color','white');
set(gca,'FontWeight','bold');
sizeScreen = get(0,'ScreenSize');
set(scene1,'position',sizeScreen);
camlight('headlight');
axis equal; grid on; box on;
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)');
view([2]);
axis([-1 6 -3 3 0 2]);  % Ajustado al rango de la función

scale = 4;  % Tamaño del robot
MobileRobot_5; % Cargar modelo del robot
H1 = MobilePlot_4(x1_square(1), y1_square(1), phi_square(1), scale); hold on;
H2 = plot3(hx_square(1), hy_square(1), 0, 'b', 'lineWidth', 2);  % Trazo en azul

% Animación del movimiento
for k = 1:N_square
    delete(H1); delete(H2);
    H1 = MobilePlot_4(x1_square(k), y1_square(k), phi_square(k), scale);
    H2 = plot3(hx_square(1:k), hy_square(1:k), zeros(1,k), 'b', 'lineWidth', 2);
    pause(ts);
end
```



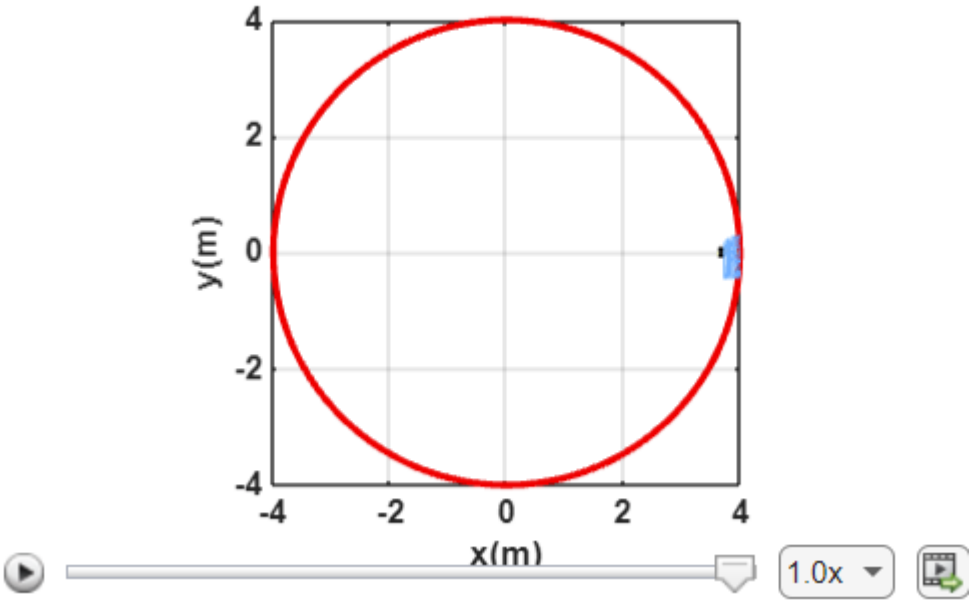
circular de radio 4 mediante las funciones paramétricas

- con un radio `r = 4` y 300 valores de θ entre 0 y 2π.
- Se calcula la orientación `phi_square` para que el robot siempre mire tangente a la circunferencia.
- Se grafica en 3D con color rojo y se anima el movimiento paso a paso.




```
clear
close all
clc
```

```
%%%%%%%%%%%%%%%%%%%%%%%%% TRAYECTORIA COMPLETA %%%%%%%%%%%%%%%%%%%%%%%%%

r = 4;  % Radio
theta = linspace(0, 2*pi, 300);  % Ángulo de 0 a 2pi (círculo completo)

hx_square = r * cos(theta);  % x(t) desplazado para que entre en escena [0 a 8]
hy_square = r * sin(theta);  % y(t) desplazado para que entre en escena [0 a 8]

%%%%%%%%%%%%%%%%%%%% ORIENTACIÓN DEL ROBOT %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

phi_square = atan2(diff(hy_square), diff(hx_square));
phi_square(end+1) = phi_square(end);  % Igualar longitudes

x1_square = hx_square;  % Para animación con robot
y1_square = hy_square;

N_square = length(theta);
ts = 0.03;  % Tiempo de muestreo

%%%%%%%%%%%%%%%%%%%%%% SIMULACIÓN GRÁFICA %%%%%%%%%%%%%%%%%%%%%%
scene1 = figure;
set(scene1,'Color','white');
set(gca,'FontWeight','bold');
sizeScreen = get(0,'ScreenSize');
set(scene1,'position',sizeScreen);
camlight('headlight');
axis equal; grid on; box on;
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)');
view([2]);
axis([-4 4 -4 4 0 2]);

scale = 4;  % Tamaño del robot
MobileRobot_5; % Cargar modelo del robot
H1 = MobilePlot_4(x1_square(1), y1_square(1), phi_square(1), scale); hold on;
H2 = plot3(hx_square(1), hy_square(1), 0, 'r', 'lineWidth', 2);

% Animación del movimiento
for k = 1:N_square
    delete(H1); delete(H2);
    H1 = MobilePlot_4(x1_square(k), y1_square(k), phi_square(k), scale);
    H2 = plot3(hx_square(1:k), hy_square(1:k), zeros(1,k), 'r', 'lineWidth', 2);
    pause(ts);
end
```

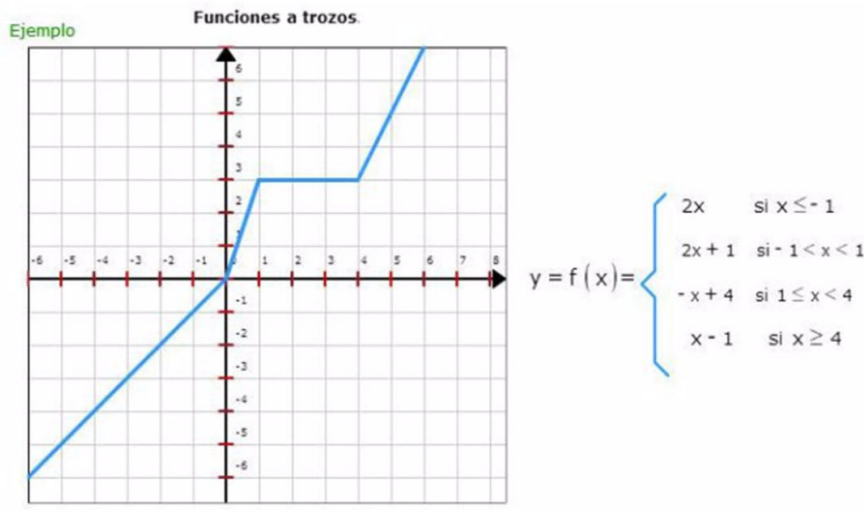


TRAYECTORIA 3

- Se define un conjunto de velocidades lineales (u) y angulares (w) para cada segundo durante 7 segundos.
- Se inicia en la posición (-6, -6) con una orientación de 45° (π/4 rad).
- Se emplea el método de Euler para calcular nuevas posiciones y orientaciones del robot en cada paso.
- Se grafica en 3D la trayectoria en color cian, junto con dos gráficas:
- Velocidad lineal (u) vs. tiempo.
- Velocidad angular (w) vs. tiempo.

Trayectoria 3:

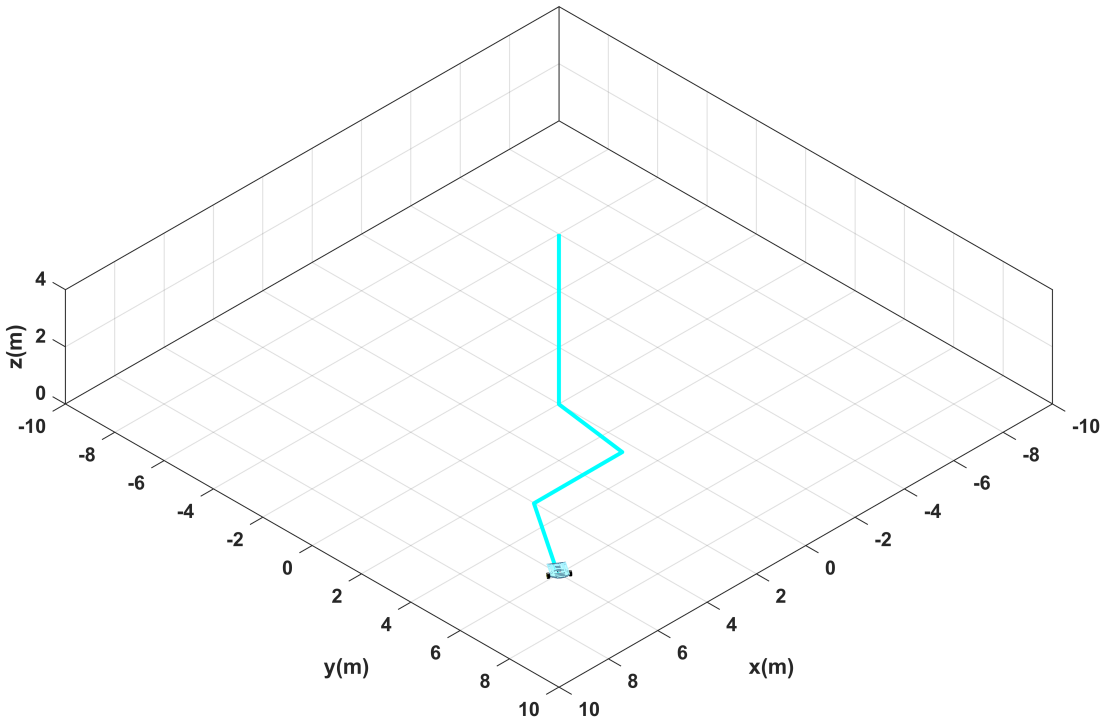
X= [-6 a 6]



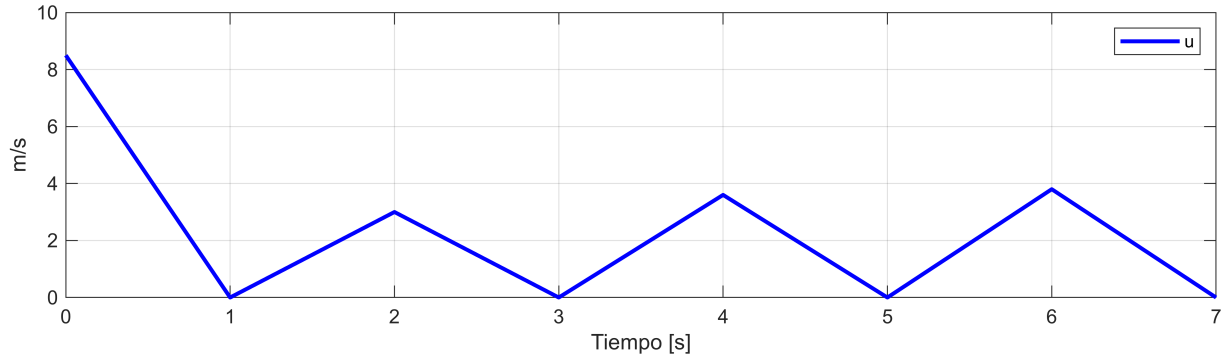
```
clear
close all
clc

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% TIEMPO %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

tf = 7;             % Tiempo de simulacion en segundos (s)
ts = 1;            % Tiempo de muestreo en segundos (s)
t = 0: ts: tf;       % Vector de tiempo
N = length(t);       % Muestras

%%%%%%%%%%%%%%%%%%%%%%%%% CONDICIONES INICIALES %%%%%%%%%%%%%%%%%%%%%%%%%%%%

x1 = zeros (1,N+1);  % Posición en el centro del eje que une las ruedas (eje x) en
metros (m)
y1 = zeros (1,N+1);  % Posición en el centro del eje que une las ruedas (eje y) en
metros (m)
phi = zeros(1, N+1); % Orientacion del robot en radianes (rad)

x1(1) = -6;    % Posicion inicial eje x
y1(1) = -6;   % Posicion inicial eje y
phi(1) = pi/4;   % Orientacion inicial del robot

%%%%%%%%%%%%%%%%%%%%%%%%%%%% PUNTO DE CONTROL %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

hx = zeros(1, N+1);  % Posicion en el punto de control (eje x) en metros (m)
hy = zeros(1, N+1);  % Posicion en el punto de control (eje y) en metros (m)

hx(1) = x1(1); % Posicion en el punto de control del robot en el eje x
hy(1) = y1(1); % Posicion en el punto de control del robot en el eje y

%%%%%%%%%%%%%%%%%%%%%%% VELOCIDADES DE REFERENCIA %%%%%%%%%%%%%%%%%%%%%%%%%%
u = [8.5,    0,  3,      0,  3.6,      0,  3.8, 0];
w = [0,   0.65,  0,  -1.43,    0,  0.972,    0, 0];

%%%%%%%%%%%%%%%%%%%%%%%%% BUCLE DE SIMULACION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for k=1:N

    phi(k+1)=phi(k)+w(k)*ts; % Integral numérica (método de Euler)

    %%%%%%%%%%%%%%%%%%%%%% MODELO CINEMATICO %%%%%%%%%%%%%%%%%%%%%%%%

    xp1=u(k)*cos(phi(k+1));
    yp1=u(k)*sin(phi(k+1));

    x1(k+1)=x1(k) + xp1*ts ; % Integral numérica (método de Euler)
    y1(k+1)=y1(k) + yp1*ts ; % Integral numérica (método de Euler)
```

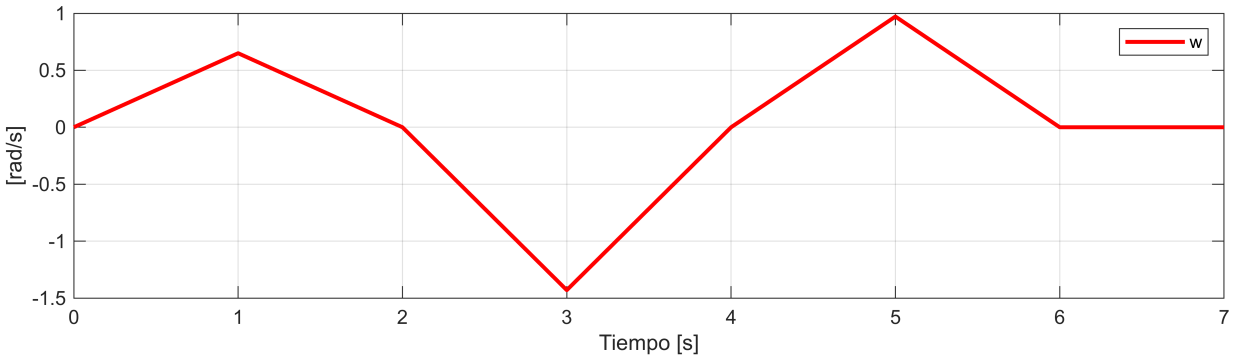
```
    % Posicion del robot con respecto al punto de control
    hx(k+1)=x1(k+1);
    hy(k+1)=y1(k+1);

end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%% SIMULACION VIRTUAL 3D %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% a) Configuracion de escena

scene=figure;  % Crear figura (Escena)
set(scene,'Color','white'); % Color del fondo de la escena
set(gca,'FontWeight','bold') ;% Negrilla en los ejes y etiquetas
sizeScreen=get(0,'ScreenSize'); % Retorna el tamaño de la pantalla del computador
set(scene,'position',sizeScreen); % Congigurar tamaño de la figura
camlight('headlight'); % Luz para la escena
axis equal; % Establece la relación de aspecto para que las unidades de datos sean
las mismas en todas las direcciones.
grid on; % Mostrar líneas de cuadrícula en los ejes
box on; % Mostrar contorno de ejes
xlabel('x(m)'); ylabel('y(m)'); zlabel('z(m)'); % Etiqueta de los eje

view([135 35]); % Orientacion de la figura
axis([-10 10 -10 10 0 4]); % Ingresar limites minimos y maximos en los ejes x y z
[minX maxX minY maxY minZ maxZ]

% b) Graficar robots en la posicion inicial
scale = 4;
MobileRobot_5;
H1=MobilePlot_4(x1(1),y1(1),phi(1),scale);hold on;

% c) Graficar Trayectorias
H2=plot3(hx(1),hy(1),0,'c','lineWidth',2);

% d) Bucle de simulacion de movimiento del robot

step=1; % pasos para simulacion

for k=1:step:N

    delete(H1);
    delete(H2);

    H1=MobilePlot_4(x1(k),y1(k),phi(k),scale);
    H2=plot3(hx(1:k),hy(1:k),zeros(1,k),'c','lineWidth',2);

    pause(ts);
```

```
end
```




```
%%%%%%%%%%%%%%%%%%%%%%%%%%%% Graficas %%%%%%%%%%%%%%%%%%%%%%%%%%%%
graph=figure;  % Crear figura (Escena)
set(graph,'position',sizeScreen); % Congigurar tamaño de la figura
subplot(211)
plot(t,u,'b','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('m/
s'),legend('u');
subplot(212)
plot(t,w,'r','LineWidth',2),grid('on'),xlabel('Tiempo [s]'),ylabel('[rad/
s]'),legend('w')
```


u
m/s
Tiempo [s]


w
[rad/s]
Tiempo [s]