

Actividad 2.2 (Parametrización de trayectorias)

Daniel Ruán Aguilar A01731921

Fredy Canseco Santos A01735589

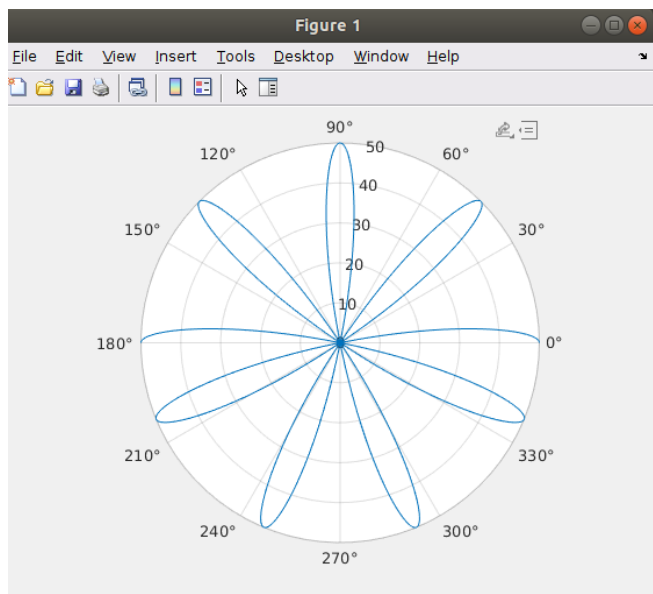
José Angel Ramírez Ramírez A01735529

1. Implementar el código requerido para generar la parametrización de las siguientes trayectorias en un plano 2D.

Trayectoria 1:

Primero se grafica la trayectoria en coordenadas polares

```
Editor - /home/ruan/Escritorio/Descargas/parametrizacion_trayectorias_polares.m
Act1_map_coord.mbx parametrizacion_trayectorias_polares.m
1 %Limpieza de pantalla
2 clear all
3 close all
4 clc
5
6 %Flor de 8 petalos
7
8 %Graficación de la trayectoria en coordenadas polares
9 %Definimos el parámetro theta como un vector en radianes
10 theta=0:pi/100:pi;
11 %Definimos el parámetro r como una función respecto a theta
12 r=50*cos(8*theta);
13 %Se ingresan parametros a la función polar para la representación gráfica (theta(radianes), r)
14 %figure(1)
15 %polarplot(theta,r)
16
17
```

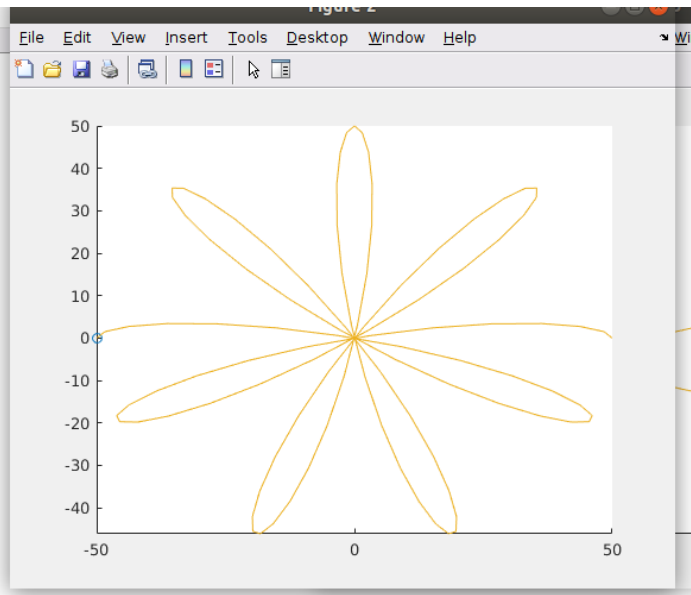


Luego se convierte la trayectoria en coordenadas cartesianas

```

17
18 %Graficación de la trayectoria en coordenadas cartesianas
19 %Transformamos las coordenadas polares a cartesianas
20 x= r.*cos(theta);
21 y= r.*sin(theta);
22
23 %graficamos la trayectoria
24 figure(2)
25 comet(x,y)
26
27

```



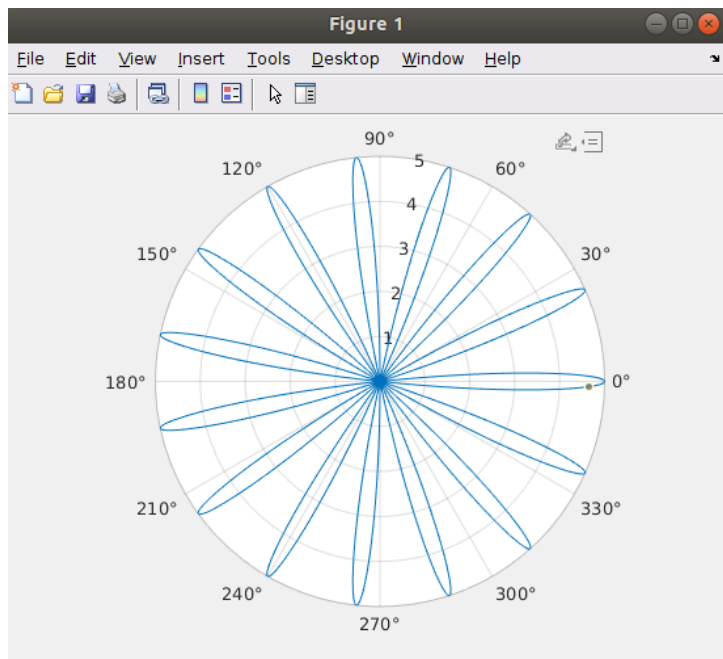
Trayectoria 2:

Primero se grafica la trayectoria en coordenadas polares

```

Act1_map_coord.mlx parametrizacion_trayectorias_polares.m
1 %Limpieza de pantalla
2 clear all
3 close all
4 clc
5
6 %Flor de 15 petalos
7
8 %Graficación de la trayectoria en coordenadas polares
9 %Definimos el parámetro theta como un vector en radianes
10 theta=0:pi/1000:pi;
11 %Definimos el parámetro r como una función respecto a theta
12 r=5*cos(15*theta);
13 %Se ingresan parametros a la función polar para la representación gráfica (theta(radianes), r)
14 %figure(1)
15 %polarplot(theta,r)
16

```

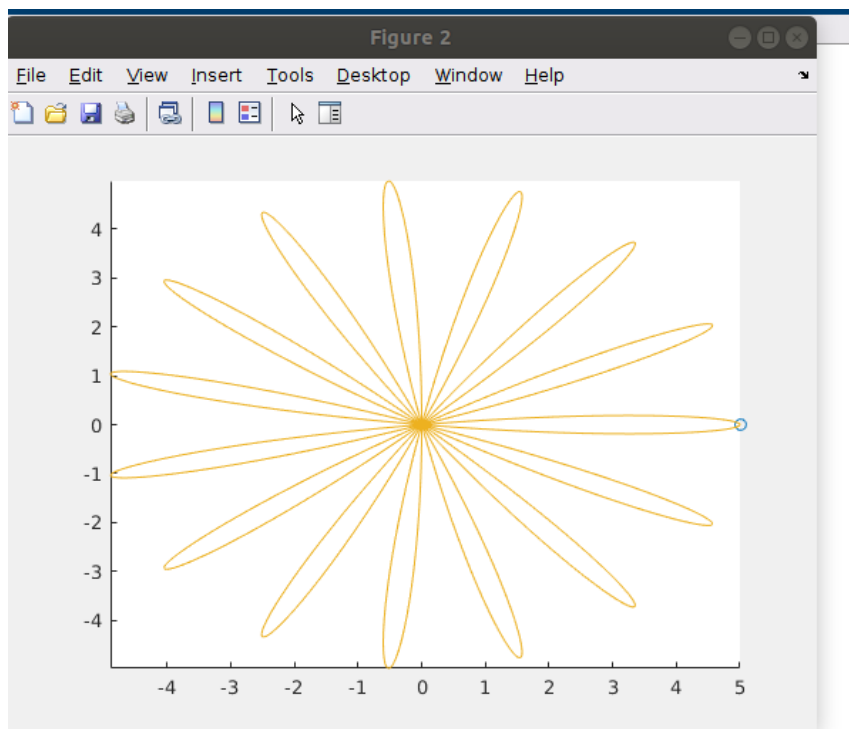


Luego se convierte la trayectoria en coordenadas cartesianas

```

17
18 %Graficación de la trayectoria en coordenadas cartesianas
19 %Trasformamos las coordenadas polares a cartesianas
20 x= r.*cos(theta);
21 y= r.*sin(theta);
22
23 %graficamos la trayectoria
24 figure(2)
25 comet(x,y)
26
27
28

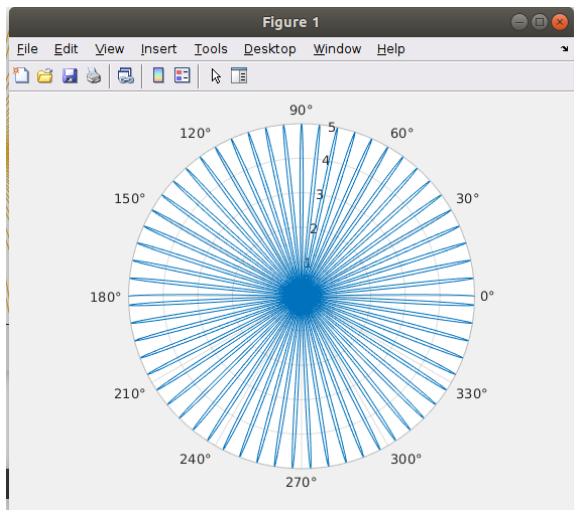
```



Trayectoria 3:

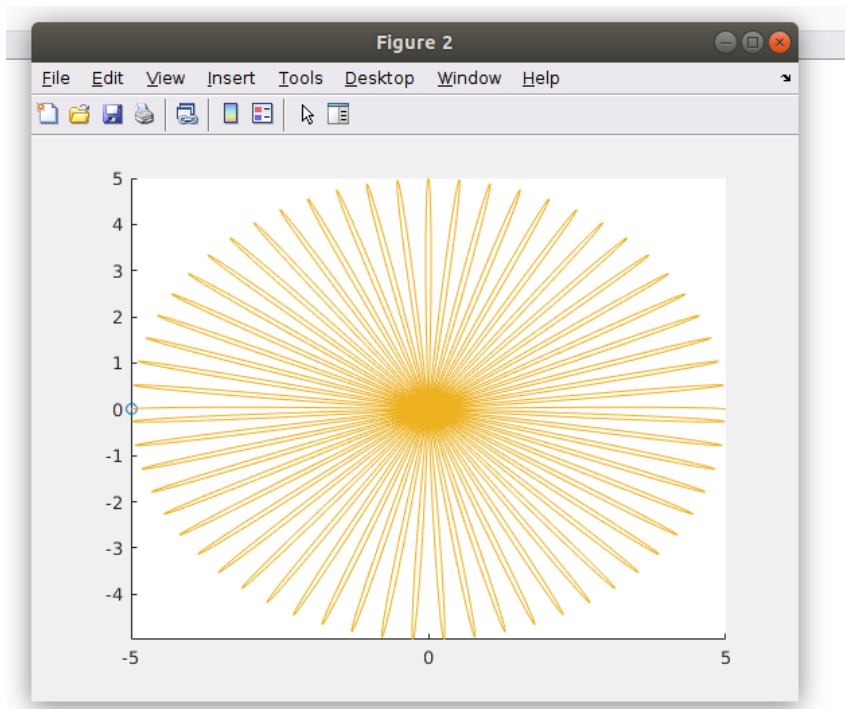
Primero se grafica la trayectoria en coordenadas polares

```
Editor - /home/ruan/Escritorio/Descargas/parametrizacion_trayectorias_polares.m
Act1_map_coord.mlx parametrizacion_trayectorias_polares.m +
1 %Limpieza de pantalla
2 clear all
3 close all
4 clc
5
6 %Flor de 15 petalos
7
8 %Graficación de la trayectoria en coordenadas polares
9 %Definimos el parámetro theta como un vector en radianes
10 theta=0:pi/1000:pi;
11 %Definimos el parámetro r como una función respecto a theta
12 r=5*cos(60*theta);
13 %Se ingresan parametros a la función polar para la representación gráfica (theta(radianes), r)
14 %figure(1)
15 %polarplot(theta,r)
16
17
```



Luego se convierte la trayectoria en coordenadas cartesianas

```
16
17
18 %Graficación de la trayectoria en coordenadas cartesianas
19 %Trasformamos las coordenadas polares a cartesianas
20 x= r.*cos(theta);
21 y= r.*sin(theta);
22
23 %graficamos la trayectoria
24 figure(2)
25 comet(x,y)
26
27
28
29
```



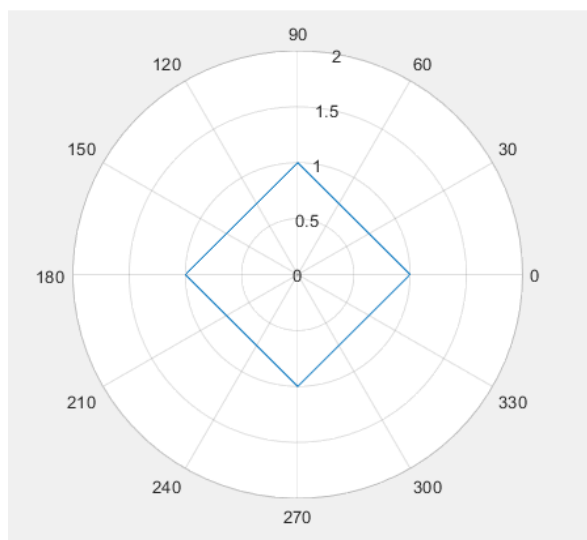
Trayectoria 4:

El primer paso es la graficación de la trayectoria como un vector en radianes en el plano polar declarando el ángulo (θ) y r (magnitud).

```
% Definimos el parámetro theta como un vector en radianes
theta = 0:pi/2:2*pi;

% Definimos el parámetro r como una función respecto a theta
r = ones(1,5);

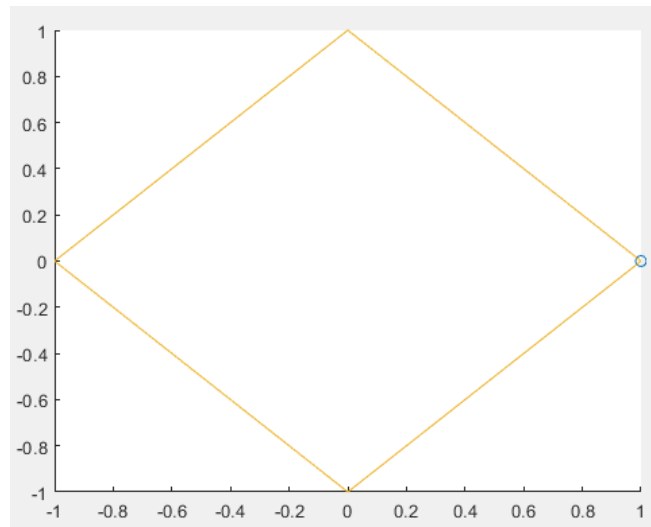
% Se ingresan parametros a la función polar para la representación gráfica (theta(radianes), r)
figure(1)
polarplot(theta,r)
```



Posteriormente se transforma las coordenadas a cartesianas con “x” y “y”, para poderlo graficar.

```
% Transformamos las coordenadas polares a cartesianas
x = r.*cos(theta);
y = r.*sin(theta);

% Graficamos la trayectoria en coordenadas cartesianas
figure(2)
comet(x,y)
```



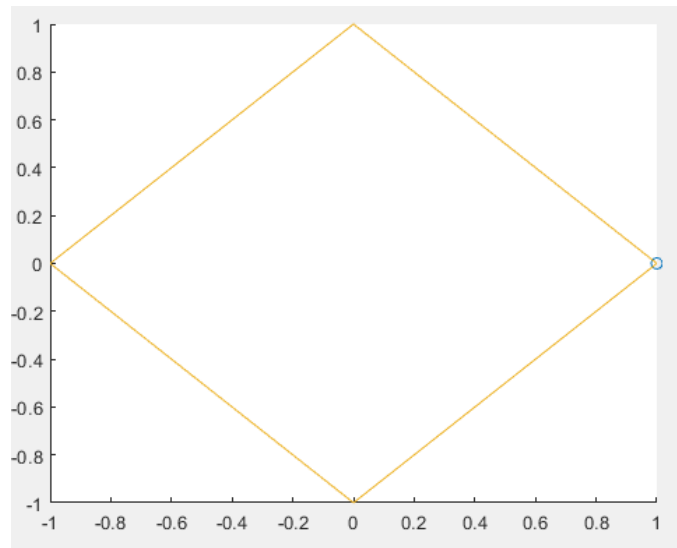
El último paso es graficar la trayectoria en coordenadas paramétricas normalizando el intervalo de tiempo y obteniendo las ecuaciones dependientes del tiempo.

```
% Graficamos la trayectoria en coordenadas paramétricas
% Se define el parámetro "t" de parametrización sobre el cual se realizará
% la proyección de trayectoria
t = [0:4];

% Se normaliza el intervalo de tiempo al intervalo de variación del ángulo
% theta
theta_t = normalize(t,"range",[0,2*pi]);

% Obtenemos las ecuaciones dependientes del tiempo
x_t = ones(1,5).*cos(theta_t);
y_t = ones(1,5).*sin(theta_t);

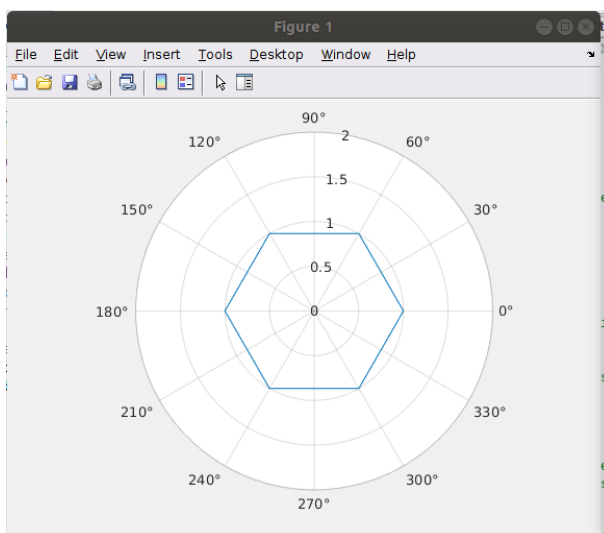
% Graficamos la trayectoria
figure(3)
comet(x_t, y_t)
```



Trayectoria 5:

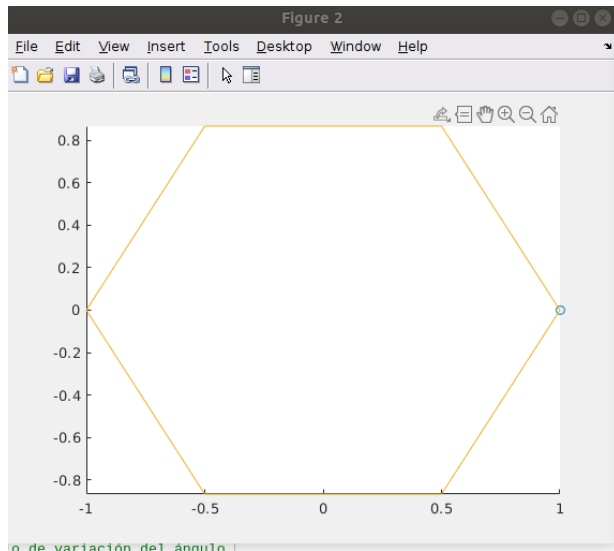
El primer paso es la graficación de la trayectoria como un vector en radianes en el plano polar declarando el ángulo (θ) y r (magnitud).

```
% Definimos el parámetro theta como un vector en radianes
theta = 0:pi/3:2*pi;
% Definimos el parámetro r como una función respecto a theta
r = ones(1,7);
% Se ingresan parametros a la función polar para la representación gráfica (theta(radianes), r)
figure(1)
polarplot(theta,r)
```



Posteriormente se transforma las coordenadas a cartesianas con “x” y “y”, para poderlo graficar.

```
% Transformamos las coordenadas polares a cartesianas
x = r.*cos(theta);
y = r.*sin(theta);
% Graficamos la trayectoria en coordenadas cartesianas
figure(2)
comet(x,y)
```

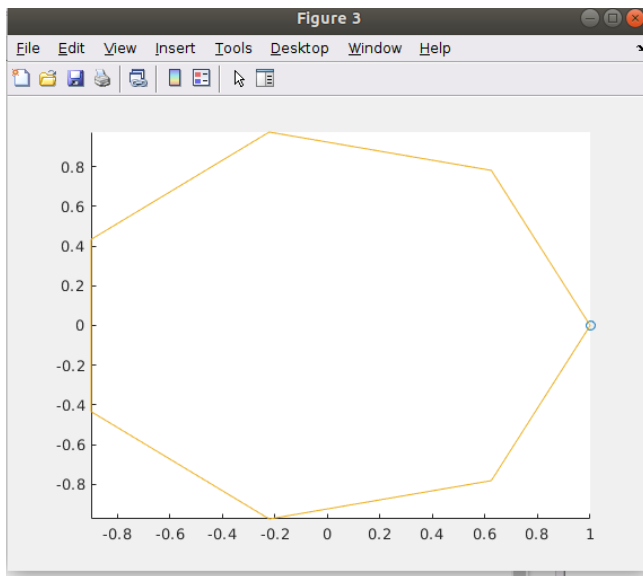


El último paso es graficar la trayectoria en coordenadas paramétricas normalizando el intervalo de tiempo y obteniendo las ecuaciones dependientes del tiempo.


```

% Graficamos la trayectoria en coordenadas paramétricas
% Se define el parámetro "t" de parametrización sobre el cual se realizará
% la proyección de trayectoria
t = 0:1:7;
% Se normaliza el intervalo de tiempo al intervalo de variación del ángulo
% theta
theta_t = normalize(t,"range",[0,2*pi]);
% Obtenemos las ecuaciones dependientes del tiempo
x_t = ones(1,8).*cos(theta_t);
y_t = ones(1,8).*sin(theta_t);
% Gramos la trayectoria
figure(3)
comet(x_t, y_t)

```



Trayectoria 6:

El primer paso es la graficación de la trayectoria como un vector en radianes en el plano polar declarando el ángulo (θ) y r (magnitud).

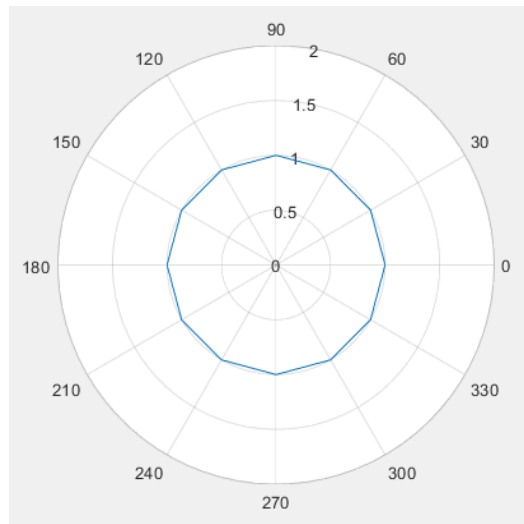
```

% Definimos el parámetro theta como un vector en radianes
theta = 0:pi/6:2*pi;

% Definimos el parámetro r como una función respecto a theta
r = ones(1,13);

% Se ingresan parametros a la función polar para la representación gráfica (theta(radianes), r)
figure(1)
polarplot(theta,r)

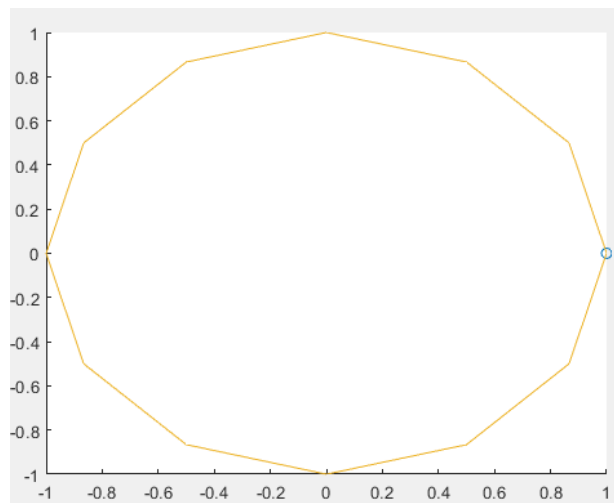
```



Posteriormente se transforma las coordenadas a cartesianas con “x” y “y”, para poderlo graficar.

```
% Transformamos las coordenadas polares a cartesianas
x = r.*cos(theta);
y = r.*sin(theta);

% Graficamos la trayectoria en coordenadas cartesianas
figure(2)
comet(x,y)
```



El último paso es graficar la trayectoria en coordenadas paramétricas normalizando el intervalo de tiempo y obteniendo las ecuaciones dependientes del tiempo.

```
% Graficamos la trayectoria en coordenadas paramétricas
% Se define el parámetro "t" de parametrización sobre el cual se realizará
% la proyección de trayectoria
t = [0:12];

% Se normaliza el intervalo de tiempo al intervalo de variación del ángulo
% theta
theta_t = normalize(t,"range",[0,2*pi]);

% Obtenemos las ecuaciones dependientes del tiempo
x_t = ones(1,13).*cos(theta_t);
y_t = ones(1,13).*sin(theta_t);

% Graficamos la trayectoria
figure(3)
comet(x_t, y_t)
```

