Actividad 1 (Mapeo de coordenadas)

Bruno Manuel Zamora Garcia A 01798275

1. Declaración de variables simbólicas:

aqui se definen x(t), y(t) y θ(t) como funciones simbólicas del tiempo t. Esto significa que, a diferencia de valores numéricos fijos, se consideran variables simbólicas que permiten hacer operaciones algebraicas y diferenciaciones.

```
% Limpieza de pantalla
clear all           % Borra todas las variables del espacio de trabajo
close all           % Cierra todas las figuras abiertas
clc                 % Limpia la ventana de comandos

tic % Inicia un temporizador para medir el tiempo de ejecución

%--------------------------------------------------------------------------
% Declaración de variables simbólicas
% x, y, th se asumen como funciones simbólicas del tiempo t
%--------------------------------------------------------------------------
syms x(t) y(t) th(t) t
```

2. Creación del vector de posición y de velocidades

- En ξinercial se agrupan las coordenadas generalizadas (x,y,θ).
- Luego obtengo las velocidades generalizadas al derivar cada componente respecto a t. Así, si ξinercial =[x(t);y(t);θ(t)], entonces ξinercial=d/dt[x(t);y(t);θ(t)].

```
%--------------------------------------------------------------------------
% Creamos el vector de posición en coordenadas inerciales
% Este vector corresponde a (x, y, th) en el espacio de estados
%--------------------------------------------------------------------------
xi_inercial = [x; y; th];
disp('Coordenadas generalizadas');
```

```
Coordenadas generalizadas
```

```
pretty(xi_inercial);
```

```
/  x(t) \
|       |
|  y(t) |
|       |
\ th(t) /
```

```


%--------------------------------------------------------------------------
% Creamos el vector de velocidades generalizadas
% Diferenciamos cada componente de xi_inercial con respecto a t
%--------------------------------------------------------------------------
```

```
xip_inercial = diff(xi_inercial, t);
disp('Velocidades generalizadas');
```

```
Velocidades generalizadas
```

```
pretty(xip_inercial);
```

```
/   d      \
|  -- x(t) |
|  dt      |
|          |
|   d      |
|  -- y(t) |
|  dt      |
|          |
|  d       |
| -- th(t) |
\ dt       /
```

3. Definición de la matriz de rotación simbólica

- P(:,:,1) guarda el vector [x;y;θ]. Lo uso como si fuera la posición en 3D, aunque la tercera componente sea θ.
- R(:,:,1) es la matriz de rotación en torno al eje Z usando el ángulo θ. Está definida simbólicamente con `cos(th)` y `sin(th)`.

```
%-------------------------------------------------------------------------
% Defino mi vector de posición y matriz de rotación (para la parte simbólica)
% P(:,:,1) = [x; y; th], que corresponde a xi_inercial
% R(:,:,1) es la matriz de rotación alrededor del eje Z usando la variable
simbólica th
%-------------------------------------------------------------------------
P(:,:,1) = [x; y; th];
R(:,:,1) = [cos(th) -sin(th)  0;
            sin(th)  cos(th)  0;
            0        0        1];
```

4. Transformación del marco inercial al local

- Esta línea hace la multiplicación entre la matriz de rotación R y el vector de posición P. En términos simbólicos, nos está mostrando cómo se transforman (x,y,θ) del marco global (inercial) al marco local.
- Aquí θ aparece también como tercera componente, así que simbólicamente se está aplicando la rotación a (x,y,θ).

```
%-------------------------------------------------------------------------
% Realizo mi transformación del marco de referencia global al local
% xi_local = R * P (en el contexto simbólico)
%-------------------------------------------------------------------------
xi_local = R(:,:,1) * P(:,:,1)
```

```
xi_local =
```

$$\begin{pmatrix} \cos(\text{th}(t))\, x(t) - \sin(\text{th}(t))\, y(t) \\ \cos(\text{th}(t))\, y(t) + \sin(\text{th}(t))\, x(t) \\ \text{th}(t) \end{pmatrix}$$

5. Conjunto de coordenadas inerciales (valores numéricos)

En esta matriz estan todas las coordenadas (x,y,θ) (en grados) que quiero mapear al marco local. Cada fila representa un inciso distinto, con un valor de x,y y θ.

```
%--------------------------------------------------------------------------
% A partir de aquí, añadimos el for para recorrer las coordenadas inerciales
% con valores numéricos en grados (x, y, θ)
%--------------------------------------------------------------------------
coordenadas_inerciales = [
   -5   9   -2;   % a)
   -3   8   63;   % b)
    5  -2   90;   % c)
    0   0  180;   % d)
   -6   3  -55;   % e)
   10  -2   45;   % f)
    9   1   88;   % g)
    5   2   33;   % h)
   -1  -1   21;   % i)
    6   4  -40;   % j)
    5   7   72;   % k)
    7   7   30;   % l)
   11  -4  360;   % m)
   20   5  270;   % n)
   10   9  345;   % ñ)
   -9  -8    8;   % o)
    1   1   60;   % p)
    3   1  -30;   % q)
   15   2  199;   % r)
  -10   0  300;   % s)
];
```

6. Bucle for para hacer la transformación

- Recorro todas las filas de la matriz "`coordenadas_inerciales`". Para cada fila:

1. Extraigo x, y y θ (en grados).
2. Convierto θ a radianes: *`th_rad = deg2rad(th_val);`*
3. Defino el vector de posición *`Pos_i = [x_val; y_val; 0];`*
4. Creo la matriz de rotación numérica *`Rot_i`* con *`cos(th_rad)`* y *`sin(th_rad).`*
5. Aplico la transformación al marco local: *`xi_local_i = Rot_i * Pos_i;`*

- Aquí solo se rotan (x,y), dejando la componente z=0.

6.Calculo la magnitud (norma) de ese vector en el marco local, es decir: $\sqrt{x_{local}^2 + y_{local}^2}$.

7.Invierto la matriz de rotación *(`inv_Rot_i`)* para comprobar que, al multiplicar, regreso a *(x,y)* en el marco inercial.

8.Imprimo resultados para verificar cada inciso.

```
% Recorremos cada conjunto (x, y, theta) de la matriz
"coordenadas_inerciales"
%--------------------------------------------------------------------------
for i = 1:size(coordenadas_inerciales,1)

    % Extraemos x, y, theta (en grados) de la fila i
    x_val  = coordenadas_inerciales(i,1);
    y_val  = coordenadas_inerciales(i,2);
    th_val = coordenadas_inerciales(i,3);

    % Convertimos theta de grados a radianes para las funciones cos() y sin()
    th_rad = deg2rad(th_val);

    % Defino el vector de posición en 3D (asumiendo la componente Z=0)
    Pos_i = [x_val; y_val; 0];

    % Construyo la matriz de rotación en torno a Z con el ángulo th_rad
    Rot_i = [cos(th_rad), -sin(th_rad), 0;
             sin(th_rad),  cos(th_rad), 0;
             0,            0,           1];

    % Transformación al marco local: multiplicación Rot_i * Pos_i
    xi_local_i = Rot_i * Pos_i;

    % Magnitud del vector en el marco local (solo se consideran x, y)
    magnitud_i = sqrt( xi_local_i(1)^2 + xi_local_i(2)^2 );

    % Transformación inversa para comprobar que se recuperan las coordenadas
originales
    inv_Rot_i = inv(Rot_i);
    xi_inercial_i = inv_Rot_i * xi_local_i;

    % Mostramos los resultados en pantalla
    fprintf('                                                    \n')
    fprintf('\n----- Inciso %d -----\n', i);
    fprintf('Coordenadas inerciales: (x = %.2f, y = %.2f, th = %.2f°)\n',
x_val, y_val, th_val);
    fprintf('xi_local_i    = [%.4f; %.4f; %.4f]\n', xi_local_i(1),
xi_local_i(2), xi_local_i(3));
    fprintf('magnitud_i    = %.4f\n', magnitud_i);
```

```
    fprintf('xi_inercial_i = [%.4f; %.4f; %.4f]\n', xi_inercial_i(1),
xi_inercial_i(2), xi_inercial_i(3));
    fprintf('-----------------------------------------------------------\n');
end
```

```
----- Inciso 1 -----
Coordenadas inerciales: (x = -5.00, y = 9.00, th = -2.00°)
xi_local_i    = [-4.6829; 9.1690; 0.0000]
magnitud_i    = 10.2956
xi_inercial_i = [-5.0000; 9.0000; 0.0000]
-----------------------------------------------------------

----- Inciso 2 -----
Coordenadas inerciales: (x = -3.00, y = 8.00, th = 63.00°)
xi_local_i    = [-8.4900; 0.9589; 0.0000]
magnitud_i    = 8.5440
xi_inercial_i = [-3.0000; 8.0000; 0.0000]
-----------------------------------------------------------

----- Inciso 3 -----
Coordenadas inerciales: (x = 5.00, y = -2.00, th = 90.00°)
xi_local_i    = [2.0000; 5.0000; 0.0000]
magnitud_i    = 5.3852
xi_inercial_i = [5.0000; -2.0000; 0.0000]
-----------------------------------------------------------

----- Inciso 4 -----
Coordenadas inerciales: (x = 0.00, y = 0.00, th = 180.00°)
xi_local_i    = [0.0000; 0.0000; 0.0000]
magnitud_i    = 0.0000
xi_inercial_i = [0.0000; 0.0000; 0.0000]
-----------------------------------------------------------

----- Inciso 5 -----
Coordenadas inerciales: (x = -6.00, y = 3.00, th = -55.00°)
xi_local_i    = [-0.9840; 6.6356; 0.0000]
magnitud_i    = 6.7082
xi_inercial_i = [-6.0000; 3.0000; 0.0000]
-----------------------------------------------------------

----- Inciso 6 -----
Coordenadas inerciales: (x = 10.00, y = -2.00, th = 45.00°)
xi_local_i    = [8.4853; 5.6569; 0.0000]
magnitud_i    = 10.1980
xi_inercial_i = [10.0000; -2.0000; 0.0000]
-----------------------------------------------------------

----- Inciso 7 -----
Coordenadas inerciales: (x = 9.00, y = 1.00, th = 88.00°)
xi_local_i    = [-0.6853; 9.0294; 0.0000]
magnitud_i    = 9.0554
xi_inercial_i = [9.0000; 1.0000; 0.0000]
-----------------------------------------------------------

----- Inciso 8 -----
Coordenadas inerciales: (x = 5.00, y = 2.00, th = 33.00°)
xi_local_i    = [3.1041; 4.4005; 0.0000]
magnitud_i    = 5.3852
xi_inercial_i = [5.0000; 2.0000; 0.0000]
-----------------------------------------------------------

----- Inciso 9 -----
```

```
Coordenadas inerciales: (x = -1.00, y = -1.00, th = 21.00°)
xi_local_i    = [-0.5752; -1.2919; 0.0000]
magnitud_i    = 1.4142
xi_inercial_i = [-1.0000; -1.0000; 0.0000]
-----------------------------------------------------

----- Inciso 10 -----
Coordenadas inerciales: (x = 6.00, y = 4.00, th = -40.00°)
xi_local_i    = [7.1674; -0.7925; 0.0000]
magnitud_i    = 7.2111
xi_inercial_i = [6.0000; 4.0000; 0.0000]
-----------------------------------------------------

----- Inciso 11 -----
Coordenadas inerciales: (x = 5.00, y = 7.00, th = 72.00°)
xi_local_i    = [-5.1123; 6.9184; 0.0000]
magnitud_i    = 8.6023
xi_inercial_i = [5.0000; 7.0000; 0.0000]
-----------------------------------------------------

----- Inciso 12 -----
Coordenadas inerciales: (x = 7.00, y = 7.00, th = 30.00°)
xi_local_i    = [2.5622; 9.5622; 0.0000]
magnitud_i    = 9.8995
xi_inercial_i = [7.0000; 7.0000; 0.0000]
-----------------------------------------------------

----- Inciso 13 -----
Coordenadas inerciales: (x = 11.00, y = -4.00, th = 360.00°)
xi_local_i    = [11.0000; -4.0000; 0.0000]
magnitud_i    = 11.7047
xi_inercial_i = [11.0000; -4.0000; 0.0000]
-----------------------------------------------------

----- Inciso 14 -----
Coordenadas inerciales: (x = 20.00, y = 5.00, th = 270.00°)
xi_local_i    = [5.0000; -20.0000; 0.0000]
magnitud_i    = 20.6155
xi_inercial_i = [20.0000; 5.0000; 0.0000]
-----------------------------------------------------

----- Inciso 15 -----
Coordenadas inerciales: (x = 10.00, y = 9.00, th = 345.00°)
xi_local_i    = [11.9886; 6.1051; 0.0000]
magnitud_i    = 13.4536
xi_inercial_i = [10.0000; 9.0000; 0.0000]
-----------------------------------------------------

----- Inciso 16 -----
Coordenadas inerciales: (x = -9.00, y = -8.00, th = 8.00°)
xi_local_i    = [-7.7990; -9.1747; 0.0000]
magnitud_i    = 12.0416
xi_inercial_i = [-9.0000; -8.0000; 0.0000]
-----------------------------------------------------

----- Inciso 17 -----
Coordenadas inerciales: (x = 1.00, y = 1.00, th = 60.00°)
xi_local_i    = [-0.3660; 1.3660; 0.0000]
magnitud_i    = 1.4142
xi_inercial_i = [1.0000; 1.0000; 0.0000]
-----------------------------------------------------

----- Inciso 18 -----
Coordenadas inerciales: (x = 3.00, y = 1.00, th = -30.00°)
```

```
xi_local_i    = [3.0981; -0.6340; 0.0000]
magnitud_i    = 3.1623
xi_inercial_i = [3.0000; 1.0000; 0.0000]
-------------------------------------------------------

----- Inciso 19 -----
Coordenadas inerciales: (x = 15.00, y = 2.00, th = 199.00°)
xi_local_i    = [-13.5316; -6.7746; 0.0000]
magnitud_i    = 15.1327
xi_inercial_i = [15.0000; 2.0000; 0.0000]
-------------------------------------------------------

----- Inciso 20 -----
Coordenadas inerciales: (x = -10.00, y = 0.00, th = 300.00°)
xi_local_i    = [-5.0000; 8.6603; 0.0000]
magnitud_i    = 10.0000
xi_inercial_i = [-10.0000; 0.0000; 0.0000]
-------------------------------------------------------
```

```
toc
```

```
Elapsed time is 0.848880 seconds.
```

Conclusión

Con estos pasos, logro generar el mapeo entre un marco de referencia inercial (x,y,θ) y un marco local, aplicando una matriz de rotación en 2D (extendida a 3x3). Primero defino todo de manera simbólica, para luego pasar a los valores numéricos y comprobar caso por caso (inciso a inciso) cómo se transforma la posición.

De esta forma, el código combina la representación simbólica (para ver la forma general de la transformación) con el cálculo numérico (para cada set de coordenadas específicas).