Projeto de linha de transmissão

Leonardo Santos - GRR20196154

A)

Para realizar essa simulação foi optado em utilizar o exemplo disponibilizado pelo Matlab.

Primeramente foram ajustados o waypoints, de forma a realizar a trajetoria retangular conforme ilustrada pela Figura 1 a seguir

```
waypoints = [0 1; 0 10; 0 20; 10 20; 20 20; 20 10; 20 0; 10 0; 0 0];
sampleTime = 0.05;              % Sample time [s]
tVec = 0:sampleTime:20;         % Time array

initPose = [waypoints(1,:)'; 0]; % Initial pose (x y theta

goalPoints = [0 ; -1] ;
goalRadius = 1;
```

Figura 1: Codigo da trajetória retangular

Realizando a simulção chegou-se no seguinte resultado:

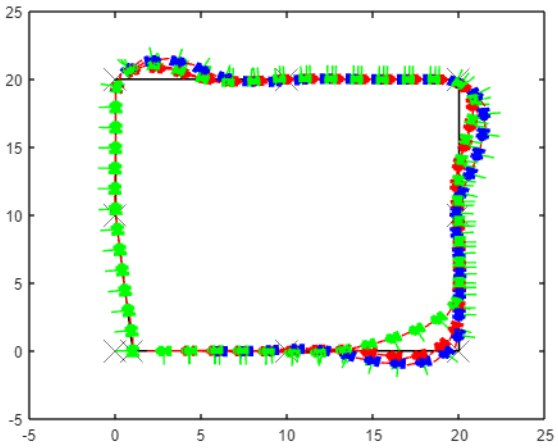

Figura 2: trajetória retangular

Após isso foi adaptado o código de forma a relizar a trajetória, o qual está ilustrado pela Figura 3 a seguir.

```
circle = 0:0.1:2*pi;
waypoints = [cos(circle') sin(circle')] * 10;
% Define the total time and the sample rate
sampleTime = 0.05;              % Sample time [s]
tVec = 0:sampleTime:20;         % Time array
initPose = [waypoints(1,:)'; pi/2]; % Initial pose (x y theta)

goalPoints = [0 ; -5] ;
goalRadius = 1;

```

Figura 3: Codigo da trajetória retangular

Em seguida foi feito a trajetoria circular do o resultado esta ilustrado pela Figura 4 a seguir.

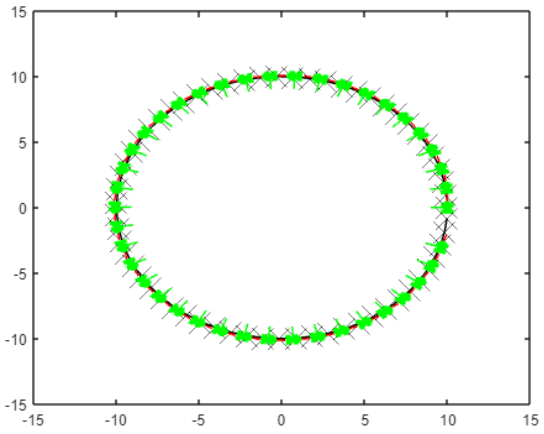

Figura 4: Trejatória Circular

B

Pode se dizer que o erro médio de cada modelo(MSE), é a distancia média de cada modelo para o poligono definido pelos waypoints. Sendo assim para fazer esse calculo foi feito um ajuste no codigo de forma o numero de amostras simuladas fosse que o mesmo numero de pontos presentes nos waypoints. Então foi calculado o erro médio. Para isso foi feito os seguintes trechos de código, que estão sendo ilustrados pelas Figura 5, Figura 6, Figura 7 a seguir,

```
i = 1
n = size(waypoints)[1]
MSEx = 0
MSEy = 0

while i < n
    MSEx_unicycle(i) = (unicyclePose(i,1) - waypoints(i,1))^2;
    MSEy_unicycle(i) = (unicyclePose(i,2) - waypoints(i,2))^2;
    MSEx = MSEx +  MSEx_unicycle(i);
    MSEy = MSEy +  MSEy_unicycle(i);
    i = i+1;
end

MSE = (MSEx+MSEy)/n
```

Figura 5: MSE retangular uniciclo

```
i = 1
n = size(waypoints)[1]
MSEx = 0
MSEy = 0

while i < n
    MSEx_bicycle(i) = (bicyclePose(i,1) - waypoints(i,1))^2;
    MSEy_bicycle(i) = (bicyclePose(i,2) - waypoints(i,2))^2;
    MSEx = MSEx +  MSEx_bicycle(i);
    MSEy = MSEy +  MSEy_bicycle(i);
    i = i+1;
end

MSE = (MSEx+MSEy)/n
```

Figura 6: MSE retangular uniciclo

```
1
2  i = 1
3  n = size(waypoints)[1]
4  MSEx = 0
5  MSEy = 0
6
7  while i < n
8      MSEx_diff(i) = (diffDrivePose(i,1) - waypoints(i,1))^2;
9      MSEy_diff(i) = (diffDrivePose(i,2) - waypoints(i,2))^2;
10     MSEx = MSEx +  MSEx_diff(i);
11     MSEy = MSEy +  MSEy_diff(i);
12     i = i+1;
13 end
14
15 MSE = (MSEx+MSEy)/n
16
```

Figura 7: MSE retangular uniciclo

E com isso chegou-se aos seguintes valore de MSE:

$$\mathrm{MSE}_{\mathrm{uni}} = 5.91$$
$$\mathrm{MSE}_{\mathrm{bi}} = 6.3$$
$$\mathrm{MSE}_{\mathrm{dif}} = 5.83$$

Portanto, podemos realizar uma suposição de que o movimento dos três modelos não estão sendo limitados pela velocidade angular maxima. Apesar dos resultadoe é visivel que o modelo diferencial é oque apresenta o melhor comportamento em curvas perpendiculares, o que é observado na Figura 2.

Em seguida foi realizado o mesmo processo para a trajtória circular, onde foi calculado MSEs muito próximos de zero, o que era de se esperar visto que a varição da direção é constante. E portanto a velocidade angular também nao influencia no desempenho de cada modelo.