

TFG_01162. Análisis de Experiencias Físicas en una Mesa Neumática mediante Visión Artificial.

ESTELA RODRÍGUEZ ARES

TUTOR: JOSÉ RICARDO PARAMO VELA

Contents

- FUNCIÓN PRINCIPAL - SEGUIMIENTO DEL MOVIMIENTO
- 1. DATOS DE ENTRADA
- 2. CÁLCULO DE LA POSICIÓN
- 3. CÁLCULO DE LA VELOCIDAD
- 4. CÁLCULO DE LA ACELERACIÓN

FUNCIÓN PRINCIPAL - SEGUIMIENTO DEL MOVIMIENTO

La función tiene como objetivo realizar el seguimiento de las posiciones que va tomando el volante a lo largo del tiempo.

```
close all; clear all; clc;
```

1. DATOS DE ENTRADA

Añadir la ubicación del video:

```
ubi = "Ubicación del archivo con su extension \nEjemplo: \nD:\\Escritorio\\TFG VISION ARTIFICIAL\\MR.mp4\n";
```

```
txt = input(ubi,"s");
```

```
video_entrada = VideoReader(txt);
```

```
video_entrada = VideoReader('D:\\Usuarios\\Estela Rodar\\Escritorio\\TFG VISION ARTIFICIAL\\PROGRAMA FINAL\\MR\\MR.mp4');
```

Añadir el número de fotogramas por segundo que aparece en los detalles de las propiedades del video.

```
prompt = "Velocidad fotograma de propiedades del video \nEjemplo: \n30\n";
```

```
fotosexseg = input(prompt);
```

```
fotosexseg = 30;
```

Añadir la frecuencia deseada entre los puntos de la gráfica de posición.

```
prompt = "Espaciado, fotogramas entre posiciones \nEjemplo: \n3 \n";
```

```
espaciado_fotogramas = input(prompt);
```

Se establecen los argumentos de entrada para poder obtener el ejecutable Publish, pero en el programa los tiene que introducir el usuario

```
espaciado_fotogramas = 3;
```

2. CÁLCULO DE LA POSICIÓN

Dado un video tomado del volante realizando un movimiento rectilíneo en la mesa neumática. Se obtiene el número de fotogramas que tiene, y se va a introducir cada fotograma en la función `posicion_imagen` para hallar el centro del volante.

```
N = video_entrada.NumFrames;
```

Se crean vectores vacíos que posteriormente alojen las posiciones de los centroides del volante a lo largo del tiempo, el área y el tiempo. También se inicializa el tiempo, y se establece el espaciado de tiempo requerido para obtener los datos del volante, en función de la calidad del video.

```
Cx = [];  
Cy = [];  
T = [];  
t = 0;  
area = [];  
  
tiempo = (1/fotoxseg)*espaciado_fotogramas;
```

Se crea un bucle for que obtenga la imagen de un fotograma cada intervalo de segundos establecido en los datos de entrada. Al hallar esta imagen la introduce en la función de posición que devuelve el centro del volante, que se irá almacenando en los vectores de posiciones.

```
for i = 1:(espaciado_fotogramas):N  
    imagen = read(video_entrada,i);  
    t = t + tiempo;  
    [C,a] = FuncionExterna_DeteccionDelVolante(imagen); % Llamada a la función  
    if(~isempty(C)) % Solo se almacena si ha detectado el volante  
        Cx = [Cx ; C(:,1)]; % Almacena la posición de las x  
        Cy = [Cy ; C(:,2)]; % Almacena la posición de las y  
        T = [T ; t];  
        area = [area ; a]; % Almacena todas las áreas del volante  
    end  
end
```

Se seleccionan las unidades de medida deseadas.

```
bucle = 1;  
while (bucle)  
    % pedir = "Introducir el número correspondiente para las unidades "+...  
    % "de medida deseadas: \n1 para metros\n2 para decímetros\n3 "+...  
    % "para centímetros\n4 para milímetros\n";  
    % uni = input(pedir);  
    %  
    % Se establece las unidades en metros para poder obtener el ejecutable  
    % Publish, pero en el programa lo tiene que introducir el usuario  
  
    uni = 1;  
    if(uni == 1)  
        radio = 0.048; % Unidades en metros  
        bucle = 0;  
    elseif(uni == 2)  
        radio = 0.48; % Unidades en decímetros  
        bucle = 0;  
    elseif(uni == 3)  
        radio = 4.8; % Unidades en centímetros  
        bucle = 0;  
    elseif(uni == 4)  
        radio = 48; % Unidades en milímetros  
        bucle = 0;  
    else  
        ERROR = 'Número introducido no válido'  
    end  
end
```

Se haya el área media para dar los resultados en unidades de medida y no en píxeles.

```
area_media = sum(area)/length(area); % (Pixel^2) Área media del volante  
area_real = radio*radio*pi; % (unidad^2) Área real de la zona verde  
equiv = sqrt(area_real)/sqrt(area_media); % (unidad/pixel) equivalencia  
desv = std(area*(equiv^2)); % Desviación típica de las áreas obtenidas en cada fotograma para hallar el error de las medidas
```

Con la equivalencia obtenida se transforman las unidades en centímetros

```
X = Cx*equiv;
Y = Cy*equiv;
Y = max(Y) - Y; % Toma la posición inicial en Y como cero
X = max(X) - X; % Toma la posición inicial en X como cero

S = [];
S = sqrt(X.*X+Y.*Y); % Se halla el desplazamiento
```

Se grafican las posiciones obtenidas, se muestran en una tabla conjunta, y se exportan a un archivo excel para poder descargarlos.

```
figure
plot(X,Y,'kx');
title('Posición Volante');
ylabel('Eje y (m)')
xlabel('Eje x (m)')
grid on;

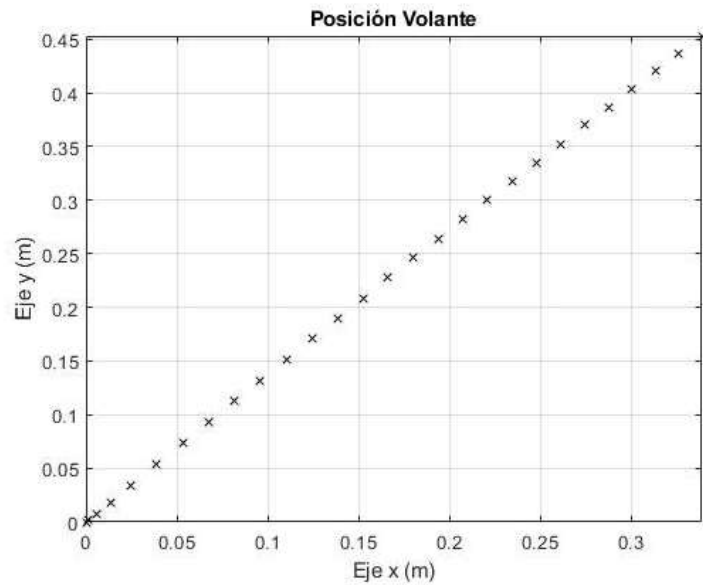
ylim([min(Y) max(Y)]);
xlim([min(X) max(X)]);

TablaC = table(T,X,Y,S)
filename = 'TablaDatosExperimentales.xlsx';
writetable(TablaC,filename,'Sheet',1,'Range','D1')
```

TablaC =

28x4 table

T	X	Y	S
0.1	0	0	0
0.2	0.0013821	0.0013245	0.0019143
0.3	0.0056132	0.0067842	0.0088053
0.4	0.013442	0.017237	0.021859
0.5	0.024671	0.03331	0.041451
0.6	0.038746	0.052868	0.065546
0.7	0.053148	0.072715	0.090068
0.8	0.067354	0.09271	0.11459
0.9	0.081526	0.11237	0.13883
1	0.095869	0.13198	0.16312
1.1	0.11014	0.15138	0.18721
1.2	0.1243	0.17065	0.21112
1.3	0.13824	0.18976	0.23478
1.4	0.15226	0.20859	0.25825
1.5	0.16616	0.22731	0.28157
1.6	0.17997	0.24567	0.30454
1.7	0.19364	0.26407	0.32746
1.8	0.20726	0.28204	0.35
1.9	0.22077	0.29998	0.37246
2	0.2343	0.31768	0.39474
2.1	0.24764	0.33524	0.41679
2.2	0.26092	0.35255	0.4386
2.3	0.27412	0.36981	0.46033
2.4	0.28718	0.38681	0.48176
2.5	0.30031	0.40358	0.50305
2.6	0.31317	0.42023	0.52409
2.7	0.32603	0.43671	0.54498
2.8	0.33863	0.45296	0.56555



3. CÁLCULO DE LA VELOCIDAD

Se calcula la velocidad experimental que va tomando el volante. Tanto en x, en y, como la absoluta. Primero los valores iniciales, y luego con un bucle for los valores que va tomando a lo largo del tiempo.

```

vx0=(X(2)-X(1))/tiempo;
Vx = [vx0; vx0];           % Añado un valor inicial porque tiene un componente menos que el vector que el de posiciones

vy0=(Y(2)-Y(1))/tiempo;
Vy = [vy0; vy0];

v0 = sqrt(vx0*vx0+vy0*vy0);
V = [v0; v0];

for i = 3:length(X)
    m=i-1;
    vx0=(X(i)-X(m))/tiempo;
    Vx = [Vx ; vx0];
    vy0=(Y(i)-Y(m))/tiempo;
    Vy = [Vy ; vy0];
    v0 = sqrt(vx0*vx0+vy0*vy0);
    vn0 = (S(i)-S(m))/tiempo;
    if (vn0<0)
        v0 = -v0;
    end
    V = [V ; v0];
end

```

Se grafican las velocidades obtenidas frente al tiempo, se muestran en una tabla conjunta, y se exporta a un archivo excel para poder descargarlos.

```

plot(T,Vx,'x');
hold on
plot(T,Vy,'x');
plot(T,V,'x');
hold off

legend('Vx','Vy','V');
title('Velocidad frente al Tiempo');
ylabel('v(m/s)')
xlabel('t(s)')
grid on;

```

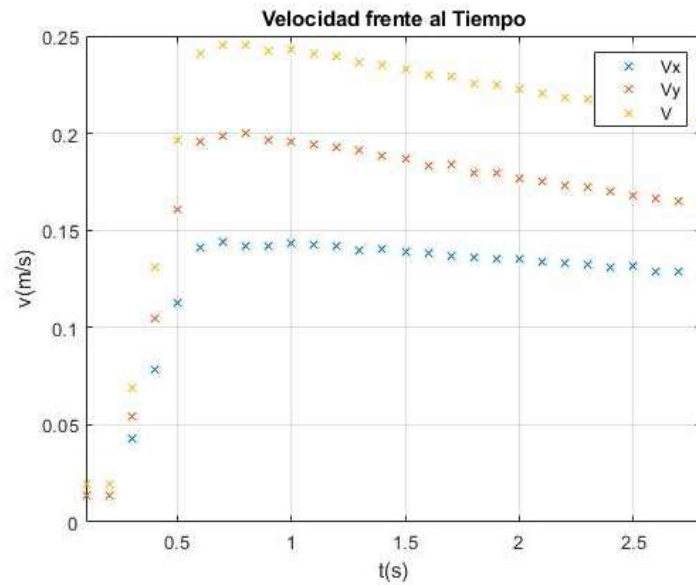
```
xlim([min(T) max(T)]);

TablaV = table(T,Vx,Vy,V)
filename = 'TablaDatosExperimentales.xlsx';
writetable(TablaV,filename,'Sheet',2,'Range','D1')
```

TablaV =

28x4 table

T	Vx	Vy	V
0.1	0.013821	0.013245	0.019143
0.2	0.013821	0.013245	0.019143
0.3	0.042311	0.054597	0.069073
0.4	0.07829	0.10453	0.1306
0.5	0.11229	0.16073	0.19607
0.6	0.14075	0.19558	0.24097
0.7	0.14402	0.19846	0.24522
0.8	0.14206	0.19995	0.24527
0.9	0.14172	0.19665	0.2424
1	0.14343	0.19602	0.24289
1.1	0.14275	0.19404	0.24089
1.2	0.14156	0.19271	0.23912
1.3	0.13945	0.19107	0.23654
1.4	0.14017	0.18837	0.2348
1.5	0.13897	0.18717	0.23313
1.6	0.13815	0.18359	0.22976
1.7	0.13668	0.18396	0.22918
1.8	0.13614	0.17973	0.22547
1.9	0.13512	0.17945	0.22463
2	0.13532	0.17696	0.22277
2.1	0.13342	0.17565	0.22058
2.2	0.13274	0.17302	0.21807
2.3	0.13206	0.17259	0.21732
2.4	0.13057	0.17	0.21436
2.5	0.13131	0.16771	0.213
2.6	0.12858	0.16653	0.21039
2.7	0.12862	0.16476	0.20902
2.8	0.12599	0.16253	0.20565



4. CÁLCULO DE LA ACELERACIÓN

Se calcula la aceleración experimental que va tomando el volante. Tanto en x, en y, como la absoluta. Primero los valores iniciales, y luego con un bucle for los valores que va tomando a lo largo del tiempo.

```
ax0=(Vx(3)-Vx(2))/tiempo;
Ax = [ax0 ; ax0; ax0];           % Añado dos valores iniciales porque tiene dos componentes menos que el vector que el de posiciones

ay0=(Vy(3)-Vy(2))/tiempo;
Ay = [ay0 ; ay0; ay0];

a0 = sqrt(ax0*ax0+ay0*ay0);
A = [a0 ; a0; a0];

for i = 4:length(Vx)
    m=i-1;
    ax0=(Vx(i)-Vx(m))/tiempo;
    Ax = [Ax ; ax0];
    ay0=(Vy(i)-Vy(m))/tiempo;
    Ay = [Ay ; ay0];
    a0 = sqrt(ax0*ax0+ay0*ay0);
    an0=(V(i)-V(m))/tiempo;
    if (an0<0)
        a0 = -a0;
    end
    A = [A ; a0];
end
```

Se grafican las aceleraciones obtenidas frente al tiempo, se muestran en una tabla conjunta, y se exporta a un archivo excel para poder descargarlos.

```
plot(T,Ax,'x');
hold on
plot(T,Ay,'x');
plot(T,A,'x');
hold off

legend('Ax','Ay','A');
title('Aceleración frente al Tiempo');
ylabel('a(m/s^2)')
xlabel('t(s)')
```

```

grid on;

xlim([min(T) max(T)]);

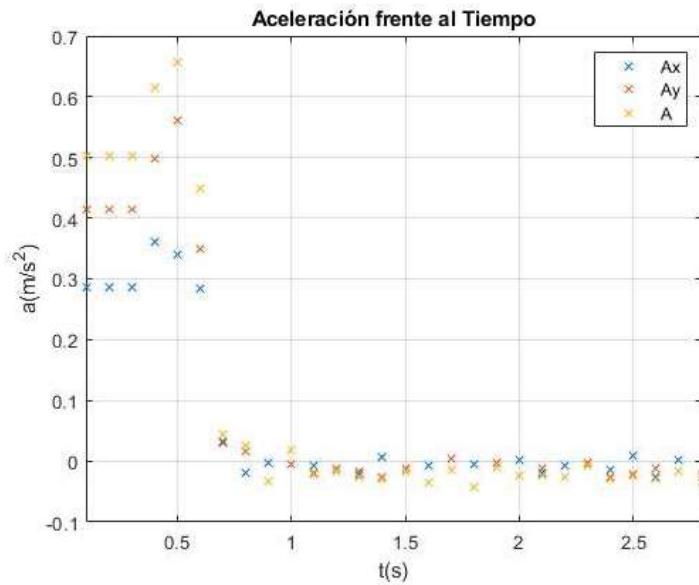
TablaA = table(T,Ax,Ay,A)
filename = 'TablaDatosExperimentales.xlsx';
writetable(TablaA,filename,'Sheet',3,'Range','D1')

```

TablaA =

28x4 table

T	Ax	Ay	A
0.1	0.2849	0.41352	0.50216
0.2	0.2849	0.41352	0.50216
0.3	0.2849	0.41352	0.50216
0.4	0.35979	0.4993	0.61543
0.5	0.33996	0.562	0.65682
0.6	0.28468	0.34857	0.45005
0.7	0.03268	0.028802	0.043561
0.8	-0.019668	0.014838	0.024637
0.9	-0.0033113	-0.032977	-0.033142
1	0.017045	-0.006343	0.018187
1.1	-0.0068187	-0.019795	-0.020937
1.2	-0.011853	-0.013285	-0.017804
1.3	-0.021161	-0.01642	-0.026784
1.4	0.0072886	-0.02699	-0.027956
1.5	-0.012	-0.01193	-0.016921
1.6	-0.0082215	-0.035876	-0.036806
1.7	-0.014707	0.0036768	-0.01516
1.8	-0.0053668	-0.042239	-0.042579
1.9	-0.010252	-0.0028358	-0.010637
2	0.0020181	-0.024878	-0.02496
2.1	-0.018968	-0.013106	-0.023055
2.2	-0.0068163	-0.026309	-0.027178
2.3	-0.0068469	-0.0042864	-0.0080779
2.4	-0.014879	-0.02586	-0.029834
2.5	0.0073491	-0.022957	-0.024105
2.6	-0.027282	-0.011775	-0.029715
2.7	0.00042767	-0.01768	-0.017685
2.8	-0.026247	-0.02232	-0.034454



A continuación se muestra una tabla con todos los datos obtenidos, y se exporta a un archivo excel para poder descargarlos.

```
Tabla = table(T, X, Y, Vx, Vy, Ax, Ay, S, V, A)
filename = 'TablaDatosExperimentales.xlsx';
writetable(Tabla,filename,'Sheet',4,'Range','D1')
```

Tabla =

28×10 table

T	X	Y	Vx	Vy	Ax	Ay	S	V	A
0.1	0	0	0.013821	0.013245	0.2849	0.41352	0	0.019143	0.50216
0.2	0.0013821	0.0013245	0.013821	0.013245	0.2849	0.41352	0.0019143	0.019143	0.50216
0.3	0.0056132	0.0067842	0.042311	0.054597	0.2849	0.41352	0.0088053	0.069073	0.50216
0.4	0.013442	0.017237	0.07829	0.10453	0.35979	0.4993	0.021859	0.1306	0.61543
0.5	0.024671	0.03331	0.11229	0.16073	0.33996	0.562	0.041451	0.19607	0.65682
0.6	0.038746	0.052868	0.14075	0.19558	0.28468	0.34857	0.065546	0.24097	0.45005
0.7	0.053148	0.072715	0.14402	0.19846	0.03268	0.028802	0.090068	0.24522	0.043561
0.8	0.067354	0.09271	0.14206	0.19995	-0.019668	0.014838	0.11459	0.24527	0.024637
0.9	0.081526	0.11237	0.14172	0.19665	-0.0033113	-0.032977	0.13883	0.2424	-0.033142
1	0.095869	0.13198	0.14343	0.19602	0.017045	-0.006343	0.16312	0.24289	0.018187
1.1	0.11014	0.15138	0.14275	0.19404	-0.0068187	-0.019795	0.18721	0.24089	-0.020937
1.2	0.1243	0.17065	0.14156	0.19271	-0.011853	-0.013285	0.21112	0.23912	-0.017804
1.3	0.13824	0.18976	0.13945	0.19107	-0.021161	-0.01642	0.23478	0.23654	-0.026784
1.4	0.15226	0.20859	0.14017	0.18837	0.0072886	-0.02699	0.25825	0.2348	-0.027956
1.5	0.16616	0.22731	0.13897	0.18717	-0.012	-0.01193	0.28157	0.23313	-0.016921
1.6	0.17997	0.24567	0.13815	0.18359	-0.0082215	-0.035876	0.30454	0.22976	-0.036806
1.7	0.19364	0.26407	0.13668	0.18396	-0.014707	0.0036768	0.32746	0.22918	-0.01516
1.8	0.20726	0.28204	0.13614	0.17973	-0.0053668	-0.042239	0.35	0.22547	-0.042579
1.9	0.22077	0.29998	0.13512	0.17945	-0.010252	-0.0028358	0.37246	0.22463	-0.010637
2	0.2343	0.31768	0.13532	0.17696	0.0020181	-0.024878	0.39474	0.22277	-0.02496
2.1	0.24764	0.33524	0.13342	0.17565	-0.018968	-0.013106	0.41679	0.22058	-0.023055
2.2	0.26092	0.35255	0.13274	0.17302	-0.0068163	-0.026309	0.4386	0.21807	-0.027178
2.3	0.27412	0.36981	0.13206	0.17259	-0.0068469	-0.0042864	0.46033	0.21732	-0.0080779
2.4	0.28718	0.38681	0.13057	0.17	-0.014879	-0.02586	0.48176	0.21436	-0.029834
2.5	0.30031	0.40358	0.13131	0.16771	0.0073491	-0.022957	0.50305	0.213	-0.024105
2.6	0.31317	0.42023	0.12858	0.16653	-0.027282	-0.011775	0.52409	0.21039	-0.029715
2.7	0.32603	0.43671	0.12862	0.16476	0.00042767	-0.01768	0.54498	0.20902	-0.017685

2.8	0.33863	0.45296	0.12599	0.16253	-0.026247	-0.02232	0.56555	0.20565	-0.034454
-----	---------	---------	---------	---------	-----------	----------	---------	---------	-----------