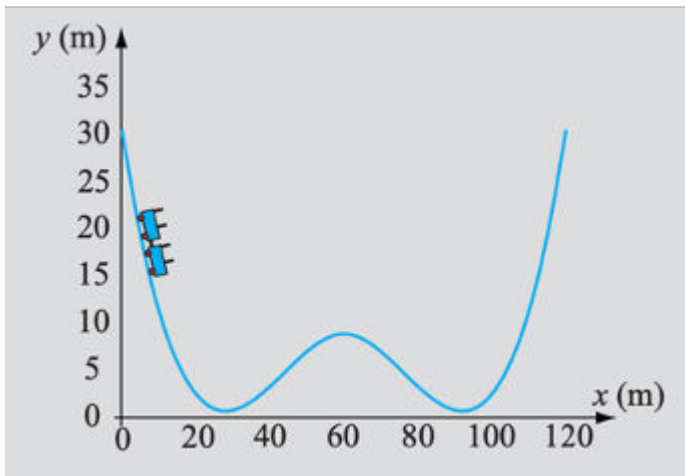


T6.GA.P02. Montaña rusa



En la siguiente tabla se da la posición (x, y) de los coches de una montaña rusa en función del tiempo t .

t (s)	0	1	2	2.5	3	3.5	4	4.5	5	5.5	6	7	8
x (m)	0	4.1	14.9	25.4	37.5	48.4	59	69.6	80.3	92.2	103.5	115.3	119.8
y (m)	31.6	22.3	7.1	2	3.54	7.6	10	8.2	4.3	1.8	5.7	21.2	31.1

La velocidad está dada por: $v = \sqrt{v_x^2 + v_y^2}$ $v_x = \frac{dx}{dt}$ $v_y = \frac{dy}{dt}$

y la aceleración:

$$a = \sqrt{a_x^2 + a_y^2} \quad a_x = \frac{d^2x}{dt^2} \quad a_y = \frac{d^2y}{dt^2}$$

a) (3p) Calcula la velocidad y la aceleración para cada instante t . Da el resultado mediante una tabla en la que las columnas sean t, x, y, v y a

b) (2p) Representa gráficamente (en la misma gráfica) v en función del tiempo mediante splines y los puntos obtenidos en el apartado b)

c) (2p) Representa gráficamente (en la misma gráfica) la aceleración a en función del tiempo mediante splines y los puntos obtenidos en el apartado b)

d) (3p) Aproxima el primer momento en que la velocidad es máxima y cuánto es dicha velocidad máxima. Para ello deriva la velocidad, representa gráficamente la derivada obtenida frente a t y realiza interpolación inversa sobre tres puntos entre los que se encuentre la raíz que buscas

Respuesta

a) Cálculo de las componentes de la velocidad v_x, v_y y v y de la aceleración a_x, a_y y a

```
clear, clc, clf
t = [0 1 2 2.5 3 3.5 4 4.5 5 5.5 6 7 8];
x = [0 4.1 14.9 25.4 37.5 48.4 59 69.6 80.3 92.2 103.5 115.3 119.8];
y = [31.6 22.3 7.1 2 3.54 7.6 10 8.2 4.3 1.8 5.7 21.2 31.1];
```

```
[vx, ax] = PySDerDes(t,x);
[vy, ay] = PySDerDes(t,y);
v = sqrt(vx.^2+vy.^2);
a = sqrt(ax.^2+ay.^2);
```

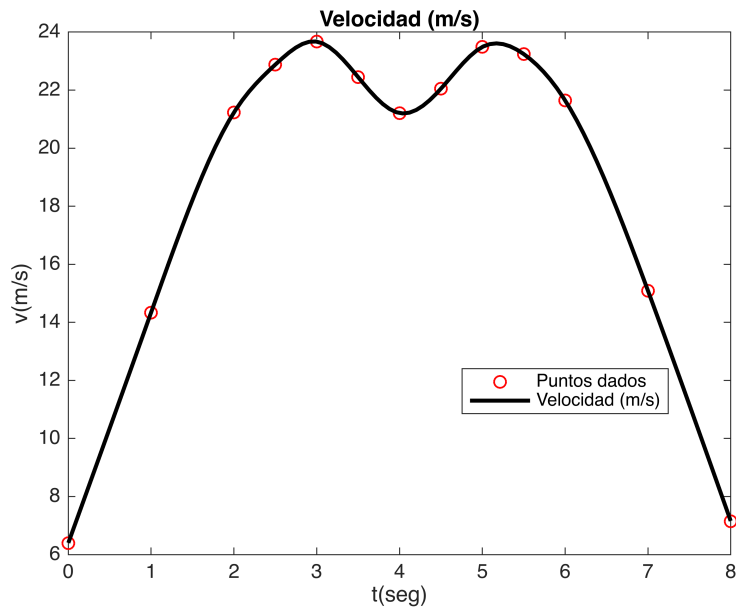
Tabla de resultados

```
T = table(t',x',y',v',a');
T.Properties.VariableNames = {'t(s)', 'x(m)', 'y(m)', 'v(m/s)', 'a(m/s^2)'};
disp(T)
```

t(s)	x(m)	y(m)	v(m/s)	a(m/s^2)
0	0	31.6	6.3941	21.039
1	4.1	22.3	14.338	8.9275
2	14.9	7.1	21.227	16.575
2.5	25.4	2	22.879	27.32
3	37.5	3.54	23.672	11.165
3.5	48.4	7.6	22.45	6.7476
4	59	10	21.208	16.8
4.5	69.6	8.2	22.049	8.4095
5	80.3	4.3	23.489	7.3756
5.5	92.2	1.8	23.242	25.712
6	103.5	5.7	21.644	20.758
7	115.3	21.2	15.09	9.2005
8	119.8	31.1	7.1507	24.67

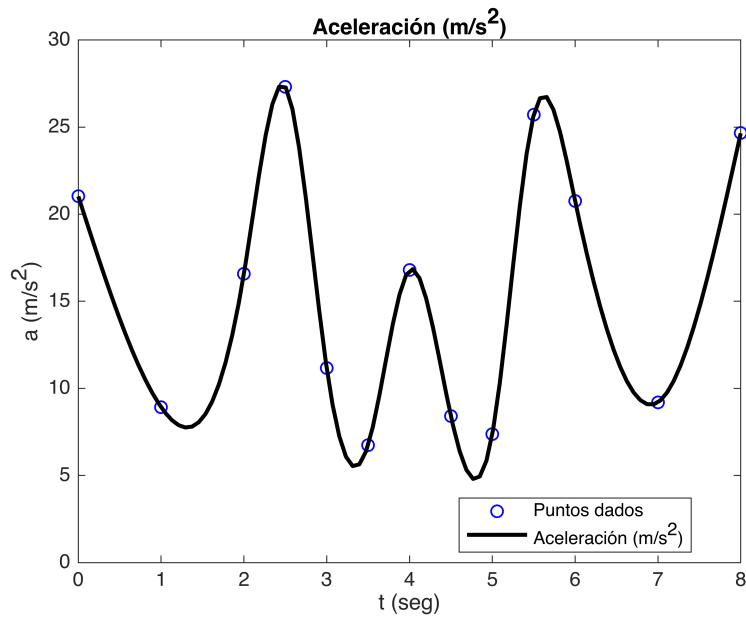
b) Dibujo de gráficas de la velocidad

```
plot(t,v,'or','LineWidth',2);
xlabel('t(seg)')
ylabel('v(m/s)')
title('Velocidad (m/s)')
hold on
tp = linspace(0.001,7.999,100);
for i = 1:100
    yp(i) = SplineCub(t,v,tp(i));
end
plot(tp,yp,'-k','LineWidth',2)
legend('Puntos dados','Velocidad (m/s)','location','best')
hold off
```



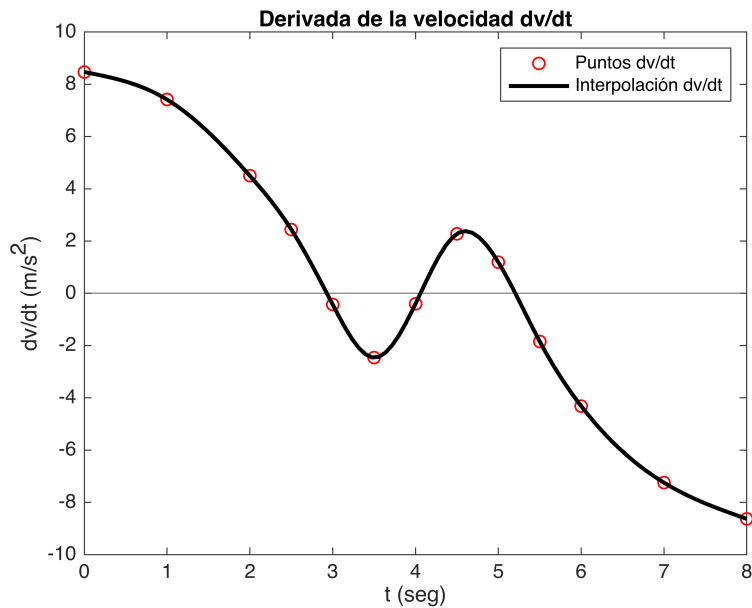
c) Dibujo de gráficas de la aceleración

```
plot(t,a,'ob','LineWidth',2)
hold on
xlabel('t (seg)')
ylabel('a (m/s^2)')
title('Aceleración (m/s^2)')
hold on
for i = 1:100
    yp(i) = SplineCub(t,a,tp(i));
end
plot(tp,yp,'-k','LineWidth',2)
legend('Puntos dados','Aceleración (m/s^2)','location','best')
hold off
```



d) Cálculo de la velocidad máxima. Derivamos la velocidad:

```
dv = PySDerDes(t,v);
plot(t,dv,'or','LineWidth',2)
hold on
for i = 1:100
    ydv(i) = SplineCub(t,dv,tp(i));
end
plot(tp,ydv,'-k','LineWidth',2)
yline(0)
xlabel('t (seg)')
ylabel('dv/dt (m/s^2)')
title('Derivada de la velocidad dv/dt')
legend('Puntos dv/dt','Interpolación dv/dt')
hold off
```



Escogemos 3 puntos próximos donde se encuentra el máximo y realizamos interpolación inversa:

```
tm = t(1,4:6);
dvm = dv(1,4:6);
vm = v(1,4:6);
tmax = NewtonINT(dvm,tm,0);
fprintf('El primer momento en que se anula la velocidad es en t = %6.4f s\n',tmax)
```

El primer momento en que se anula la velocidad es en t = 2.9100 s

```
vmax = SplineCub(t,v,tmax);
fprintf('La velocidad máxima es de %6.4f m/s\n',vmax)
```

La velocidad máxima es de 23.6660 m/s

También podríamos hacer interpolación inversa sobre todos los puntos (t, dv) mediante splines cúbicas obteniendo un resultado diferente:

```
tmax = SplineCub(dv,t,0)
```

tmax = 2.7261