



<b>Obtención de ecuación de movimiento con regresión</b>
--

<b>PRACTICA</b>	<b># 5</b>
<b>FECHA</b>	<b>28/09/2023</b>

## 1. Objetivo

En este trabajo, utilizaremos el software MATLAB para analizar el movimiento de una partícula a partir de los datos de velocidad que se nos proporcionan en el intervalo de 0 a 20 segundos. A partir de estos datos, realizaremos un regresion y obtendremos una ecuación de movimiento que describa la posición de la partícula en función del tiempo. Luego, graficaremos la posición, la velocidad y la aceleración de la partícula usando MATLAB y comentaremos los resultados obtenidos.

## 2. Marco Teórico

- Regresión no lineal:

La regresión no lineal es una técnica estadística que se utiliza para modelar la relación entre una variable dependiente y una o más variables independientes cuando esta relación no es lineal. En este caso, estamos tratando de modelar la posición de una partícula en función del tiempo a partir de los datos de velocidad proporcionados. Dado que la relación entre la posición y el tiempo no es lineal, se necesita una regresión no lineal para encontrar una función que describa con precisión el movimiento de la partícula.

- Ecuación de movimiento:

El objetivo principal de la regresión no lineal es obtener una ecuación de movimiento que relacione la posición de la partícula con el tiempo. Esta ecuación permitirá predecir la posición de la partícula en cualquier momento dentro del intervalo de tiempo especificado.

- Software MATLAB:

MATLAB es un software ampliamente utilizado en el campo de la ingeniería y la ciencia para realizar análisis numéricos y resolver problemas matemáticos. Proporciona herramientas y funciones poderosas para llevar a cabo análisis de regresión no lineal, ajustando modelos no lineales a datos experimentales.



### 3. Formulación

Para definir la función en la que realizamos la regresión usamos el siguiente comando:

inline( 'Función', 'Variable 1' , 'Variable 2');

Para realizar la regresión no lineal utilizamos el siguiente comando:

nlinfit( 'variable independiente' , 'Datos de variable dependiente' , 'Función de referencia' , 'Numero de coeficientes de función' );

### 4. Implementación en MATLAB y Resultados

#### CODIGO

```
time = 0:20;
velocity = [0,1,4,8,14,21,28,35,43,51,58,64,69,73,75,75,73,68,60,49,35];

% Adaptamos los datos a una ecuacion de 2do grado

f2 = inline('x(1)*t.^2+x(2)*t+x(3)', 'x', 't');
x2 = nlinfit(time,velocity,f2,[0 0 0]);
v2 = x2(1)*time.^2 + x2(2)*time + x2(3);

% No se adapto bien, por lo que adaptamos los datos a una ecuacion de 3er
grado

f3 = inline('x(1)*t.^3+x(2)*t.^2+x(3)*t+x(4)', 'x', 't');
x3 = nlinfit(time,velocity,f3,[0 0 0 0]);
v3 = x3(1)*time.^3+x3(2)*time.^2+x3(3)*time+x3(4);

% Encaja perfectto por lo que ya tenemos la ecuacion

syms t;
v = x3(1)*t.^3+x3(2)*t.^2+x3(3)*t+x3(4);

% Calculamos funcion posicion y aceleracion

s = int(v,t);

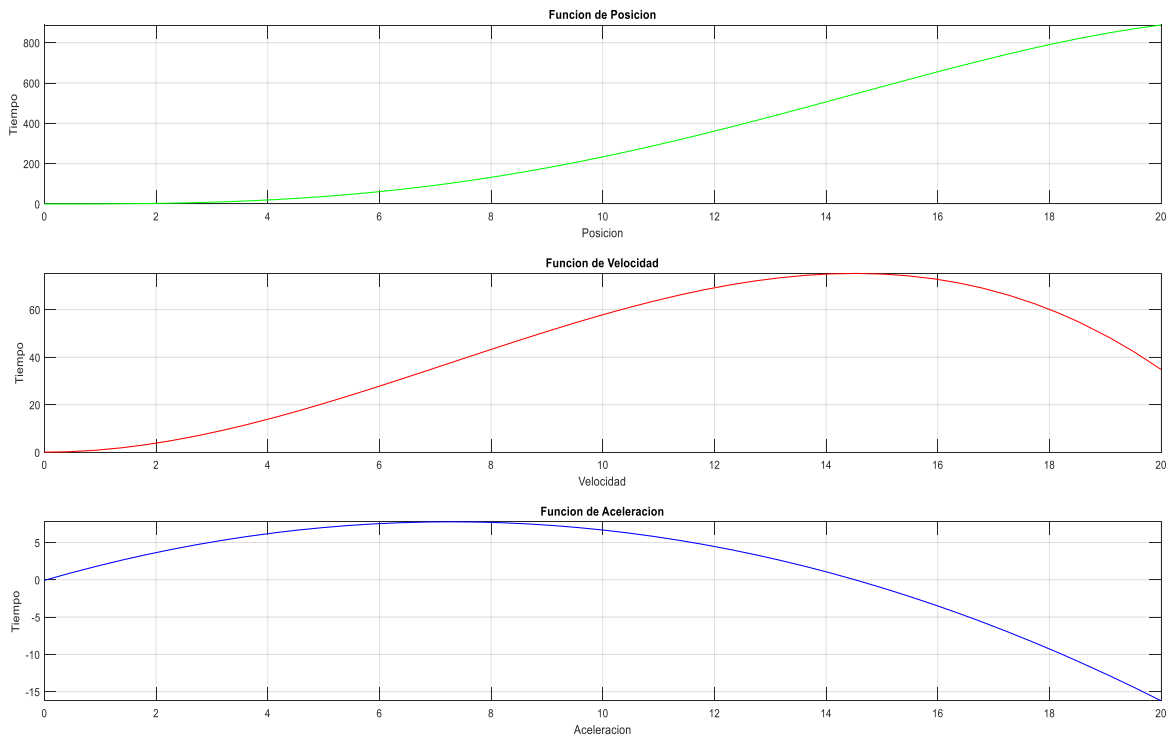
a = diff(v,t);

% Graficamos

subplot(3,1,1);
```



```
fplot(s, [0 20], 'g');  
hold on;  
grid on;  
title('Funcion de Posicion');  
xlabel('Posicion');  
ylabel('Tiempo');  
  
subplot(3,1,2);  
fplot(v, [0 20], 'r');  
hold on;  
grid on;  
title('Funcion de Velocidad');  
xlabel('Velocidad');  
ylabel('Tiempo');  
  
subplot(3,1,3);  
fplot(a, [0 20], 'b');  
hold on;  
grid on;  
title('Funcion de Aceleracion');  
xlabel('Aceleracion');  
ylabel('Tiempo');
```





Con la función de MATLAB pudimos obtener una función prácticamente perfecta para los datos de velocidad que nos brindaron, y a partir de esta pudimos obtener la ecuación de posición y velocidad.

## 5. Conclusión

Se cumplió el objetivo de realizar la regresión lineal a partir de los datos recibidos, me asombra la capacidad del software MATLAB y como realiza tantos cálculos de manera instantánea. Cada vez estoy mas familiarizado con el software y he aprendido más funciones.